Multi-Category Mesh Reconstruction From Image Collections

Alessandro Simoni*, Stefano Pini*, Roberto Vezzani, Rita Cucchiara
Department of Engineering "Enzo Ferrari", University of Modena and Reggio Emilia (Italy)
{alessandro.simoni, s.pini, roberto.vezzani, rita.cucchiara}@unimore.it

Abstract

Recently, learning frameworks have shown the capability of inferring the accurate shape, pose, and texture of an object from a single RGB image. However, current methods are trained on image collections of a single category in order to exploit specific priors, and they often make use of category-specific 3D templates. In this paper, we present an alternative approach that infers the textured mesh of objects combining a series of deformable 3D models and a set of instance-specific deformation, pose, and texture. Differently from previous works, our method is trained with images of multiple object categories using only foreground masks and rough camera poses as supervision. Without specific 3D templates, the framework learns category-level models which are deformed to recover the 3D shape of the depicted object. The instance-specific deformations are predicted independently for each vertex of the learned 3D mesh, enabling the dynamic subdivision of the mesh during the training process. Experiments show that the proposed framework can distinguish between different object categories and learn category-specific shape priors in an unsupervised manner. Predicted shapes are smooth and can leverage from multiple steps of subdivision during the training process, obtaining comparable or state-of-the-art results on two public datasets. Models and code are publicly released [1].

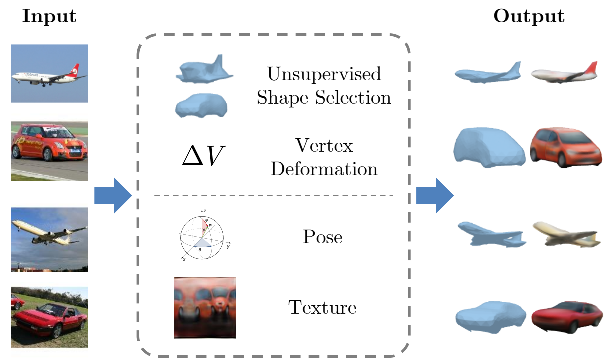

Figure 1: Overview of the proposed approach. The method predicts realistic 3D textured shapes of objects of different categories and their 3D pose from a single RGB image.

1. Introduction

In recent years, the inference of 3D object shapes from 2D images has shown astonishing progress in the computer vision community. By addressing the task as an inverse graphics problem, *i.e.* considering the 2D image as the rendering of a 3D model, several methods [16, 7, 47] have shown that deep models are capable of restoring the shape, pose, and texture of the portrayed object. While previous methods rely on direct 3D supervision [3, 6, 50, 55] or multiple views [45, 9, 48, 29], recent approaches only require segmentation masks, object keypoints, and coarse camera poses [16, 7, 47]. In the last couple of years, some methods have lessened the dependency on keypoints [47] and even on the camera viewpoint [7]. All these methods share the same underlying approach: a deep model learns a mean 3D shape, called *meanshape*, for the object category during training; then, instance-specific deformation, texture and camera pose are predicted and applied to the learned meanshape to regress the 3D model of the object.

A major limitation of existing methods is that they are category-specific: they must be trained and evaluated on image collections of a single object category. This choice has been motivated by the need of category-specific priors in order to recover the 3D shape from 2D images, which is indeed an ill-posed problem unless additional constraints are taken into account. Moreover, most of the approaches [16, 7, 47] initialize the learnable meanshape with a category-specific representative 3D model. To the best of our knowledge, there have been no attempts to extend these methods to scenarios where image collections of multiple categories are available both in training and at inference time.

In this paper, we present a multi-category approach that learns to infer the 3D mesh of an object from a single RGB image. As illustrated in Figure 1, the method learns a series

[1] https://github.com/aimagelab/mcmr
* Equal contribution.

of deformable 3D models and predicts a set of instance-specific deformation, pose, and texture based on the input image. Differently from previous approaches, the proposed framework is trained with images of multiple object categories using only foreground masks and rough camera poses as supervision. While rough camera poses could depend on the object category, this is not strictly needed for classes that share semantic keypoints. The method learns several 3D models in an unsupervised manner, *i.e.* without explicit category supervision, starting from a set of spheres and automatically selects the proper one during inference. Moreover, the instance-specific deformation is inferred by a network that independently predicts the displacement of each vertex of the learned 3D mesh, given the 3D position of the vertex and conditioned on the selected shape and the visual features extracted from the input image. The predicted deformation is naturally smooth and the number of vertices and triangles of the 3D mesh can be dynamically changed during training, with either a global or a local subdivision.

To showcase the quality of the proposed method, we present a variety of experiments in different settings on two datasets, namely Pascal3D+ [54] and CUB [49], and run several ablation studies. For instance, we test the method on multiple object categories related to the automotive environment of the Pascal3D+ dataset (*i.e.* bicycle, bus, car, and motorbike) and on the entire set of Pascal3D+ categories. Qualitative and quantitative results confirm the quality of the proposed approach and show that the model is capable of learning category-specific shape priors without direct supervision.

To sum up, our main contributions are as follows:

- We present an approach that recovers the 3D shape, pose, and texture of an object from a 2D image. The method is trained using image collections with foreground masks and coarse camera poses, but no explicit category nor 3D supervision.
- Our multi-category framework learns to distinguish between different object categories and produces meaningful meanshapes starting from a set of 3D spheres.
- Our approach predicts single vertex deformations, resulting in smooth 3D surfaces and enabling the dynamic subdivision of the learned meshes.

2. Related Work

In the last decade, many methods have been proposed to tackle the task of 3D reconstruction from a single image. However, the majority of these methods require supervisory signals which are hard to obtain in the real world and in the wild, such as 3D models [3, 6, 58, 33, 50, 40, 55, 1, 26] or multi-view image collections [45, 56, 9, 52, 48, 46, 15, 29].

Approach	**Supervision**			W/o 3D	Multi	Dynamic
	Keypoint	Camera	Mask	Template	category	subdiv.
CSDM [17]	✖	✖	✖			
CMR [16]	✖	✖	✖			
VPL [18]		✖	✖			
CSM [24]			✖			
A-CSM [23]			✖			
IMR [47]			✖			
U-CMR [7]			✖			
UMR [28]			✖	✔		
Ours		✖	✖	✔	✔	✔

Table 1: Comparison between available approaches based on training supervision, independence from offline-computed 3D templates, multi-category and dynamic subdivision support.

Recently, thanks to the development of several differentiable renderers [31, 20, 34, 30, 2], a handful of methods [17, 13, 16] have shown that the task can be addressed as an inverse graphics problem using fewer supervisory signals, such as 2D segmentation masks and object keypoints. Following methods have even relaxed these constraints, training without keypoint supervision [2, 19, 18] or known camera poses [47, 7, 28]. However, these methods require image collections of a single object category and some of them need a meaningful initialization of a category-specific shape. Differently, our method is capable of jointly learning shapes of several object categories using only foreground masks and coarse camera poses as supervision.

Another group of works that exploit differentiable renderers address the reconstruction task as a canonical surface mapping [24, 23] or a surface estimation task [25]. These methods usually require 3D supervision [25] or category-specific shape templates [24, 23]. In this paper, we focus on the 3D mesh reconstruction from single-view images without any category-specific template.

Recently, Li *et al.* [27] proposed a video-based method and the use of multiple meanshapes (referred as "base shapes") that are combined to produce a single deformable shape. This is the most similar work to our approach, but it has some key differences. Firstly, the meanshapes are defined offline and set before training, thus they are not learned. Then, they are introduced for one single dataset to exclusively cover the intra-class variation. On the contrary, our meanshapes are learned during training without category supervision and our approach can deal with several object categories and their intra- and inter-class variations.

A comparative study of literature methods is proposed in Table 1, highlighting the differences in terms of training supervision, independence from offline-computed 3D templates, multi-category and dynamic subdivision support. As shown, the proposed method still relies on camera supervi-

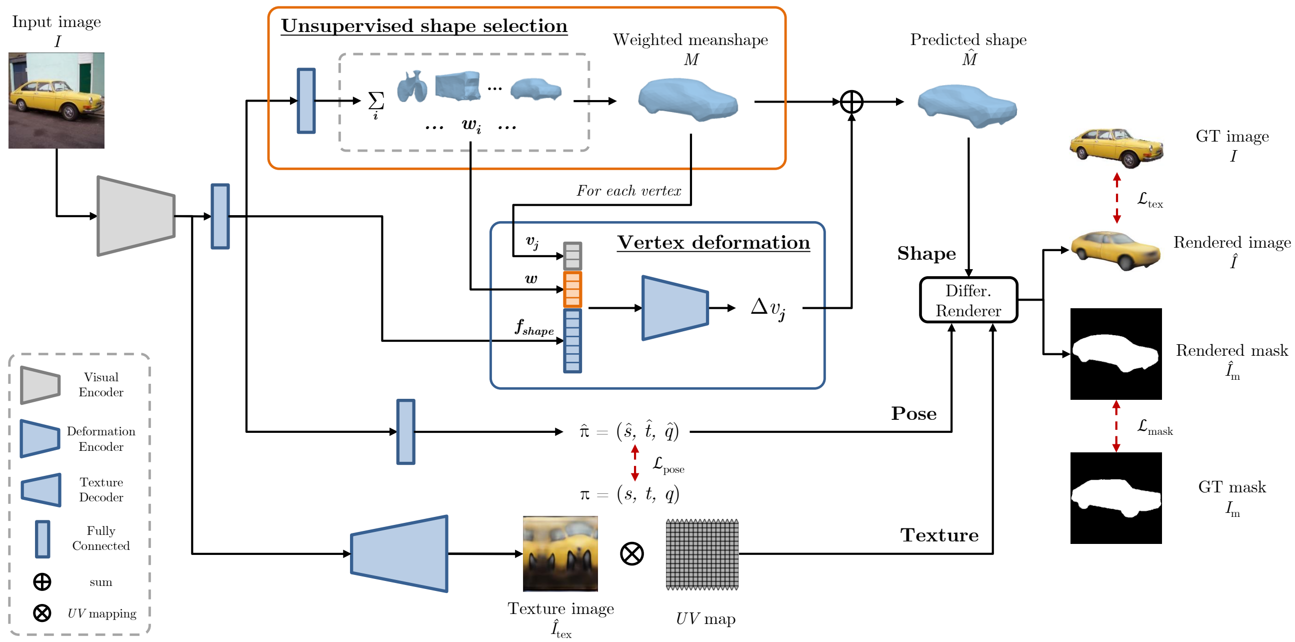

Figure 2: Overview of the proposed method. The *unsupervised shape selection* module predicts the category meanshape while the *vertex deformation* module infers the instance-specific deformation, obtaining the predicted shape. In parallel, pose and texture are estimated and then provided, along with the shape, to a differentiable renderer that renders the textured image.

sion, but introduces some unique features. Indeed, it learns category-specific shape priors in an unsupervised manner and instance-specific deformations from multi-category image collections. Moreover, the method exploits multiple steps of subdivision during the training process.

3. Method

In this section, we present the components of our method, from the input image to the reconstructed 3D textured mesh. The architecture is illustrated in Figure 2.

3.1. Preliminary definitions

Shape. As other approaches in the literature [16, 18, 7, 28], we use the triangle mesh as 3D shape representation, which is defined by a set of vertices $V = \{v_j = [x, y, z],\ j = [1, \dots, k]\}$ and a set of triangle faces F. The faces determine the connectivity between vertices, but are also related to the texture mapping. In our approach, we leverage this connectivity property and dynamically change, during training, the number of vertices and faces of the 3D shape aiming for smoothness and better textures. We refer to this technique as *dynamic mesh subdivision*.

Texture. The triangle mesh texture is represented by a texture image I_{tex} and a color map UV which maps between the 2D coordinate space of I_{tex} and the 3D coordinate space of the mesh surface of a sphere. Thus, the UV mapping is defined by spherical coordinates.

Pose. We use a weak-perspective camera projection to define the 3D object pose, as commonly done in literature. This geometric projection is a simplified version of the standard perspective projection. Thus, the object pose is parametrized by a scale factor $s \in \mathbb{R}$, a translation $t = (x, y)$ in image coordinates and a quaternion rotation q obtained by a rotation matrix computed from Euler angles (*i.e.* azimuth, elevation and roll). We define $\pi = (s, t, q)$ as the weak-perspective camera projection.

Rendering. In order to render a 3D shape with its texture, we rely on the differentiable renderer Soft Rasterizer [30]. It takes a triangle mesh, a texture image I_{tex} and an object pose π as input and outputs the rendering of the textured object as the RGB image $\hat{I}$ and the foreground mask $\hat{I}_{\text{m}}$.

3.2. Multi-category mesh reconstruction

In this paper, we aim to recover the 3D shape of an object from a single image. In the literature, this task has been often addressed by splitting it in two parts: on the one hand, the definition or learning of a category-specific base shape, named *meanshape*; on the other hand, the prediction of an instance-specific deformation of the learned shape. Differently from the majority of previous works (see Table 1), we do not need a category-specific initialization of these shapes and propose the joint and unsupervised training of shapes for multiple object categories. In the following, we provide the details of our approach.

Feature extraction. Given an RGB image $I \in \mathbb{R}^{3 \times w \times h}$

as input, the first step of our framework is the extraction of visual features with a convolutional encoder (*e.g*. ResNet-18 [12] in our experiments). These features are defined as f_{tex} and used to estimate the 3D object texture with a specific decoder. The same features are flattened and mapped into a compact version f_{shape}, used to recover the shape and its viewpoint.

Unsupervised shape selection. In contrast to current literature approaches, which are category specific, we propose an unsupervised technique that automatically learns to distinguish between different object categories. Instead of a single meanshape, we define a set of N deformable spheres and use a network to select the instance-specific meanshape according to the input image. The features f_{shape} are passed through a set of fully connected layers and a softmax function. Then, the resulting scores are used to compute a weighted sum of the mesh vertices and obtain a single mesh, approximating the argmax function over the N meanshapes. While the meanshapes are initially defined as spheres, they are updated during the training process and progressively specialize in different object categories. Formally, let $M_i = (V_i, F)$ be one of the N meanshapes and $\mathbf{w} = [w_1, \dots, w_N]$ be the output of the network. The weighted meanshape M is computed as:

$$M = (V, F) = (\sum_{i=1}^{N} w_i V_i \, , \, F) \tag{1}$$

This mesh M will be deformed according to the object depicted in the input image I, as explained in the following.

Vertex deformation. Inspired by previous works [8, 36], we develop a lightweight network which deforms the meanshape M taking as input the features f_{shape} and the 3D coordinates of a single meanshape vertex v_j at a time. We further condition the output on the selected meanshape giving the weighting scores $\mathbf{w}$ produced by the previous module as additional input. In this way, we enforce the connection between the weighted meanshape M and the predicted deformation. The module outputs a 3D displacement or deformation Δv_j of the vertex v_j in the 3D space. This approach makes the architecture independent of the number of vertices of the mesh, enabling us to predict the deformation of meshes of variable sizes. Given a set of deformations ΔV for each vertex of a meanshape M, the predicted shape can be defined as $\hat{M} = M + \Delta V = (V + \Delta V, F)$.

Dynamic mesh subdivision. In order to improve the smoothness of the predicted deformed shape, we apply during training a *dynamic subdivision* of the triangle mesh. In particular, we use a global subdivision that divides each triangle of a mesh M in 4 equal parts. Other methods that make use of mesh subdivision (*e.g*. [50, 26]) need architectural changes that drastically increase the required memory and the inference time. On the contrary, our method is not heavily affected by the mesh subdivision operation and does not require any architectural changes, thanks to the per-vertex prediction of the deformation network.

3D pose regression. We further predict the object viewpoint with a supervised regression technique using two fully connected layers which take as input the features f_{shape} and output a 3D weak-perspective pose $\hat{\pi} = (\hat{s}, \hat{t}, \hat{q})$.

Texture prediction. In order to produce a realistic 3D shape, we finally predict the texture that the differentiable renderer applies to the predicted deformed mesh $\hat{M}$. Similar to the work of Goel *et al*. [7], we use a convolutional decoder that takes as input the visual features f_{tex}, which preserve the spatiality, and directly outputs an RGB image $\hat{I}_{\text{tex}}$. The texture is mapped onto the UV space of the shape, which is homeomorphic to a sphere, so that it can be exploited by the renderer to produce the final image $\hat{I}$.

3.3. Losses and priors

The shape prediction is supervised only by two annotated information that are the binary object mask I_{m} and the 3D camera pose π.

We first handle the shape deformation applying a mask loss $\mathcal{L}_{\text{mask}} = ||I_{\text{m}} - \hat{I}_{\text{m}}||_2^2$ where $\hat{I}_{\text{m}}$ is the binary object mask produced by the renderer using the ground truth pose π. In addition to this loss, we also use some priors in order to maintain a certain smoothness of the object surface. The first prior is a laplacian smoothing loss $\mathcal{L}_{\text{smooth}} = ||LV||_2$ where the Laplace-Beltrami operator [43] minimizes the mean curvature; we apply this smoothing prior both to the predicted deformations ΔV and the vertices of the deformed shape $\hat{M}$. The second prior is a regularization term $\mathcal{L}_{\text{def}} = ||\Delta V||_2$ which prevents the network from learning large deformations and helps to produce more realistic meanshapes. Our final shape loss is represented by:

$$\mathcal{L}_{\text{shape}} = \mathcal{L}_{\text{mask}} + \mathcal{L}_{\text{smooth}} + \mathcal{L}_{\text{def}} \tag{2}$$

For the pose regression module we use a loss defined as:

$$\mathcal{L}_{\text{pose}} = ||\hat{s} - s||_2^2 + ||\hat{t} - t||_2^2 + (1 - |q * (\hat{q} \odot -\hat{q})|) \tag{3}$$

where the first two terms consist of the mean squared error for scale and translation and the last term is the geodesic quaternion loss. The operator $*$ is the Hamilton product and $\odot$ the concatenation between the original quaternion and its version rotated by 360 degrees, representing the same rotation. Moreover, following the approach proposed by Pavllo *et al*. [39], we further regularize the quaternion prediction with the penalty term $\mathcal{L}_{\text{pose\_reg}} = w^2 + x^2 + y^2 + z^2 - 1^2$ that forces the quaternion to have unit length and thus representing a valid rotation. The overall camera loss is set as:

$$\mathcal{L}_{\text{cam}} = \mathcal{L}_{\text{pose}} + \mathcal{L}_{\text{pose\_reg}} \tag{4}$$

In order to produce realistic colors and details for the object texture, we convert the rendered RGB image and the masked input image to the LAB color space and apply the following losses: a color loss $\mathcal{L}_{\text{color}} = ||\hat{I}_{ab} - (I \cdot I_{\text{m}})_{ab}||_2^2$ on the AB channels for more faithful texture details and a style loss $\mathcal{L}_{\text{style}} = ||\hat{I}_L - (I \cdot I_{\text{m}})_L||_2^2$ on the L channel for sharper high-frequency details. Moreover, we apply a perceptual loss $\mathcal{L}_{\text{percept}} = F_{\text{dist}}(\hat{I}, I \cdot I_{\text{m}})$ where F_{dist} is the metric defined by Zhang *et al.* [57] using a VGG16 backbone as feature extractor. The final texture loss is defined by:

$$\mathcal{L}_{\text{tex}} = \mathcal{L}_{\text{color}} + \mathcal{L}_{\text{style}} + \mathcal{L}_{\text{percept}} \quad (5)$$

The overall objective applied during training is a weighted sum of the shape, camera, and texture losses, obtaining a balanced learning of the different network modules. For more details about the loss weights, please refer to the supplementary material.

4. Experiments

In this section, we firstly present the employed datasets and the experimental setting. Then, we present quantitative and qualitative evaluations of our approach in comparison with literature methods. Finally, we report an ablation study on the key elements of the proposed approach.

4.1. Datasets and Experimental Setting

Two common datasets, namely Pascal3D+ [54] and CUB-200-2011 [49], have been used to evaluate the proposed approach on a diverse set of object categories and, at the same time, to obtain a comparison with the current state-of-the-art methods. As done in previous works [16, 7], 2D image collections, foreground masks and coarse camera/object poses – manually or automatically annotated – are used for training. We do not take advantage of annotated keypoint positions nor coarse 3D model correspondences.

Pascal3D+. The Pascal3D+ dataset [54] contains images of 12 object classes, from both PASCAL VOC [5, 10] and ImageNet [4], associated with 3D category-level models and coarse viewpoints [44, 35, 41, 42]. Manually-annotated foreground masks are available for the PASCAL VOC subset, while an off-the-shelf segmentation algorithm [11] is used for the other subset, as done in previous works [16, 7, 47]. We evaluate the system using the same train/test split and categories, *i.e.* *aeroplane* and *car*, of the competitors. In addition, we use the segmentation masks obtained by the novel PointRend architecture [22] and evaluate our model on a set of automotive classes, *i.e.* *bicycle, bus, car, motorbike*, and on the entire set of 12 classes in the ablation study.

CUB. We also use the images of 200 bird species and their foreground masks provided in CUB-200-2011 [49] and the camera poses computed by Kanazawa *et al.* [16], as done in previous works [16, 7, 47]. The dataset also contains 312 binary attribute labels divided in several categories.

Network architecture. Our model is composed of 5 modules: (i) a visual encoder, defined as a pre-trained ResNet-18, with an additional convolutional layer, (ii) an unsupervised shape selection module composed of two fully connected layers and a softmax activation function, (iii) a vertex deformation network with four 512-dimensional fully connected layers with random dropout and a tanh activation function, (iv) a camera pose regressor with two fully connected layers and random dropout, and (v) a texture decoder that follows the implementation of the SPADE architecture [37] with 6 upsampling steps. Additional details are available in the supplementary material.

Training procedure. We train our network on both datasets for 500 epochs with an initial learning rate of $1e^{-4}$. The meanshapes are initialized as icospheres with 162 vertices and 320 faces (corresponding to the subdivision level 3). After 350 epochs, we apply the dynamic subdivision to the 3D shapes (roughly obtaining the subdivision level 4) and reduce the learning rate to $1e^{-5}$. Our final 3D shape has roughly the same number of vertices and faces as the competitor approaches [16, 7] which use a deformable template with subdivision level fixed to 4.

All input images are cropped using the object bounding box and resized to a dimension of 256×256 and the model predicts a texture image of the same size. As data augmentation, we apply standard random jittering on the bounding box size and location and random horizontal image flipping. In addition, instead of forcing the shape to be symmetric with post-processing steps (as done in other works, *e.g.* [16, 7, 28]), we force the network to predict symmetric shapes with the following approach, similar to what is done in the work of Wu *et al.* [53]. During training, the predicted shape (*i.e.* its pose) is randomly rotated by 180 degrees around the vertical axis and compared with the flipped versions of the ground truth image and mask. In this way, the network is forced to predict symmetric shapes (along the vertical axis) and thus to consistently minimize the losses without computational overhead.

We use a batch size of 16 and Adam [21] as optimizer with a momentum of 0.9. The code is developed using the PyTorch [38] framework.

4.2. Results

In this section, we provide a thorough comparison between the proposed method and the competitors on the two previously presented datasets, Pascal3D+ and CUB.

Pascal3D+. We show the results of our method compared to the state of the art on the Pascal3D+ dataset in Table 2, using the 3D IoU metric as proposed by Tulsiani *et al.* [48]. We present two different versions of our method. Firstly, we employ the same approach used by competitors: train

Approach	Training	Aeroplane	Car	Avg
CSDM [17]	indep.	0.400	0.600	0.500
DRC [48]	indep.	0.420	0.670	0.545
CMR [16]	indep.	**0.460**	0.640	0.550
IMR [47]	indep.	0.440	0.660	0.550
U-CMR [7]	indep.	-	0.646	-
Ours (N meanshapes)	indep.	**0.460**	**0.684**	**0.572**
Ours (2 meanshapes)	**joint**	**0.448**	**0.686**	**0.567**

Table 2: 3D IoU on Pascal3D+ dataset [54]. Our method is trained on aeroplanes and cars independently using N meanshapes (one for each subclass) or on aeroplanes and cars jointly with 2 meanshapes.

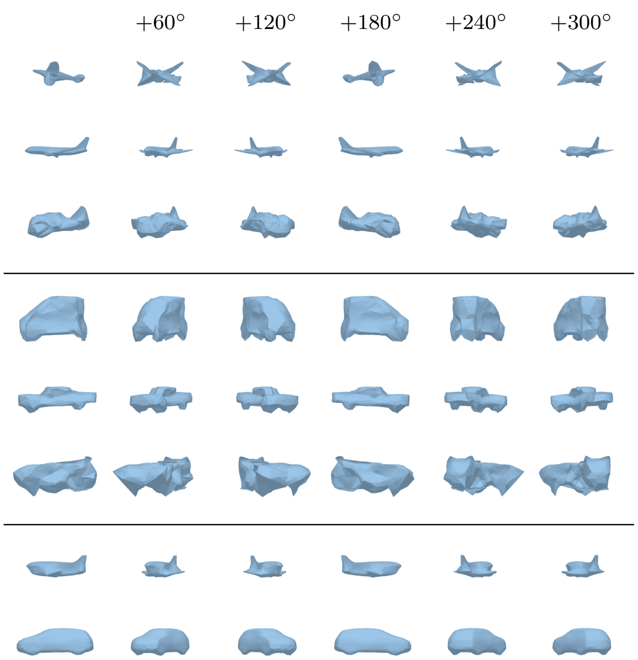

Figure 3: Some of the meanshapes learned during training on Pascal3D+. First group: aeroplane class (8 meanshapes); second group: car class (10 meanshapes); third group: aeroplane and car classes (2 meanshapes).

a different model for each class of Pascal3D+ (experiments marked as "independent training"). In this case, we set the number of meanshapes equal to the number of subclasses of Pascal3D+, *i.e.* $N = 8$ for the aeroplane class, $N = 10$ for the car class. As reported in the second-to-last row of Table 2, our method can leverage the use of multiple meanshapes and the dynamic subdivision obtaining state-of-the-art results on this dataset. In addition, we jointly train our method on both the aeroplane and the car classes using 2 meanshapes, and letting the network distinguish between the two classes. Even in this more complex scenario, we obtain comparable or state-of-the-art scores on both classes (see last row of Table 2). The learned meanshapes for these

Approach	Mask IoU ↑		Texture metrics		
	Pred cam	GT cam	SSIM ↑	L1 ↓	FID ↓
CMR [16]	**0.706**	0.734	**0.718**	**0.063**	290.32
DIB-R [2]	-	**0.757**	-	-	-
U-CMR [7]	0.637	-	0.689	0.077	**190.35**
Ours (1 meanshape)	0.658	0.721	0.717	0.064	227.24
Ours (14 meanshapes)	0.642	0.723	0.715	0.065	231.95

Table 3: Mask IoU and texture metrics on CUB dataset [49]. Our method is trained using 1 or 14 meanshapes.

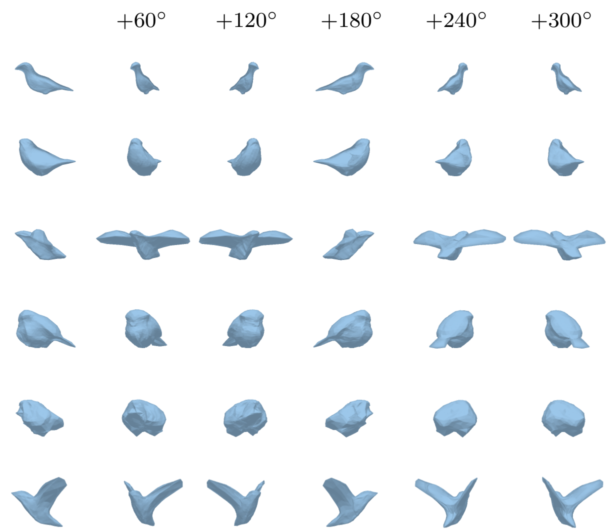

Figure 4: Some of the meanshapes learned during training on the CUB dataset using our method initialized with 14 spherical meanshapes.

three experiments, *i.e.* training on aeroplanes, on cars, and on aeroplanes and cars jointly, are shown in Figure 3. We observe that the set of meanshapes on the single classes contains both recognizable and less explainable shapes (Figure 3, top and middle): we refer the reader to the supplementary material for an analysis of the impact of the learned shapes on the weighted meanshape. On the other hand, the two meanshapes learned in an unsupervised manner using images of aeroplanes and cars correspond to these two classes (Figure 3, bottom). We show qualitative results of the joint setting on aeroplanes and cars in Figure 6 (second block).

CUB. We also evaluate our method on the CUB dataset. Results in terms of foreground mask IoU and texture metrics (SSIM [51], L1, and FID [14, 32]) are reported in Table 3. Differently from the previous case, the CUB dataset does not have a clear subdivision in classes and literature approaches have only tested on the whole dataset. Thus, we test our method in two different settings. On the one hand, we evaluate the use of a single meanshape (as done by competitors). On the other hand, we test our method

Training classes	Number of meanshapes	3D IoU ↑	Mask IoU ↑		Texture metrics		
			Pred cam	GT cam	SSIM ↑	L1 ↓	FID ↓
aeroplane, car	1	0.532	0.592	0.689	0.736	0.066	365.01
aeroplane, car	2	**0.552**	**0.671**	**0.702**	**0.737**	**0.062**	**344.80**
bicycle, bus, car, motorbike	1	0.517	0.665	0.751	0.601	0.100	390.41
bicycle, bus, car, motorbike	4	**0.543**	**0.711**	**0.759**	**0.607**	**0.094**	**380.15**
12 Pascal3D+ classes	1	0.409	0.602	0.670	0.660	0.088	357.51
12 Pascal3D+ classes	12	**0.425**	**0.620**	**0.685**	**0.665**	**0.086**	**345.90**

Table 4: Ablation study comparing the usage of several meanshapes (our proposal) against a single meanshape (as a baseline) on Pascal3D+ dataset [54] using segmentation masks obtained with PointRend [22].

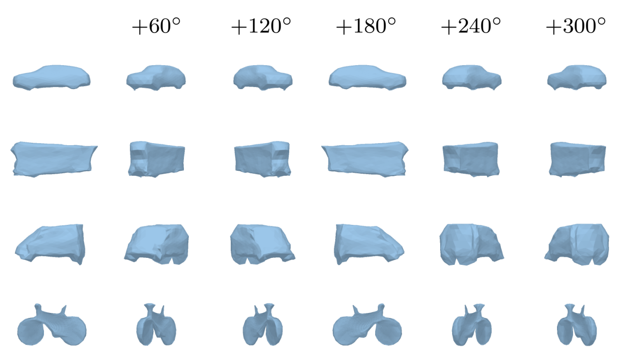

Figure 5: Meanshapes learned during training on the classes bicycle, bus, car, motorbike of the Pascal3D+ dataset [54].

initializing N deformable meanshapes, as done in previous experiments. We empirically set $N = 14$, which is equal to the number of different values of the annotated categorical attribute “has_shape”, and refer the reader to the supplementary material for an analysis of using different numbers of meanshapes on the CUB dataset. As shown, even if this dataset does contain objects of the same class “bird”, our method obtains comparable results with respect to literature approaches, on both shape and texture metrics. Even if the experiment with multiple shapes does not seem to increase the overall scores, it produces a set of insightful meanshapes learned in an unsupervised manner, as shown in Figure 4. Qualitative results are reported in Figure 6 (first block) and in the supplementary material.

4.3. Ablation study

In this section, we investigate the impact of using one or multiple meanshapes. In addition, we evaluate the influence of the dynamic subdivision approach compared to the static one. In these experiments, we use the Pascal3D+ dataset and extract precise foreground masks with PointRend [22]. Additional ablation studies and qualitative results are available in the supplementary material.

Unsupervised shape selection. As our first analysis, we evaluate the impact of the proposed unsupervised shape

Subdivision level	Mask IoU ↑		Texture metrics		
	Pred cam	GT cam	SSIM ↑	L1 ↓	FID ↓
3	0.701	**0.759**	0.600	0.096	395.96
4	0.685	0.756	0.593	0.101	385.68
$3 \rightarrow 4$	**0.711**	**0.759**	**0.607**	**0.094**	**380.15**

Table 5: Ablation study comparing different subdivision levels on Pascal3D+ dataset [54]. Model trained on 4 classes (bicycle, bus, car, motorbike) using 4 meanshapes.

selection, which enables the training with multiple meanshapes and classes. We test three different training settings using the following object categories: (i) *aeroplane*, *car*, (ii) *bicycle*, *bus*, *car*, *motorbike*, (iii) all the 12 Pascal3D+ classes. Each setting has been tested using both a single meanshape or a set of N meanshapes, in order to verify the contribution of the usage of multiple learnable shapes and their unsupervised selection. The obtained results are reported in Table 4 in terms of 3D IoU, foreground mask IoU and texture metrics. Our approach with multiple meanshapes provide the best results in all the experimental settings. Furthermore, the meanshapes learned with the four-category setting are depicted in Figure 5. Even if the meanshapes do not exactly correspond to the four classes (*e.g.*, the motorbike is missing), the meanshapes are meaningful and represent different object categories. Qualitative results are shown in Figure 6. In the supplementary material, we further evaluate the average usage of each learned meanshape throughout the test set and the classification accuracy of the unsupervised shape selection module when used as a category classifier.

Dynamic mesh subdivision. We evaluate the contribution of the dynamic mesh subdivision during the training process using the four automotive classes. We compare three different settings of the 3D mesh connectivity, in terms of icosphere subdivision level: (i) level set to 3, (ii) level set to 4, and (iii) dynamic subdivision starting from level 3 and going up to level 4. Results are reported in Table 5. As shown, the method can converge to good results even using a fixed subdivision level. However, a higher level does not

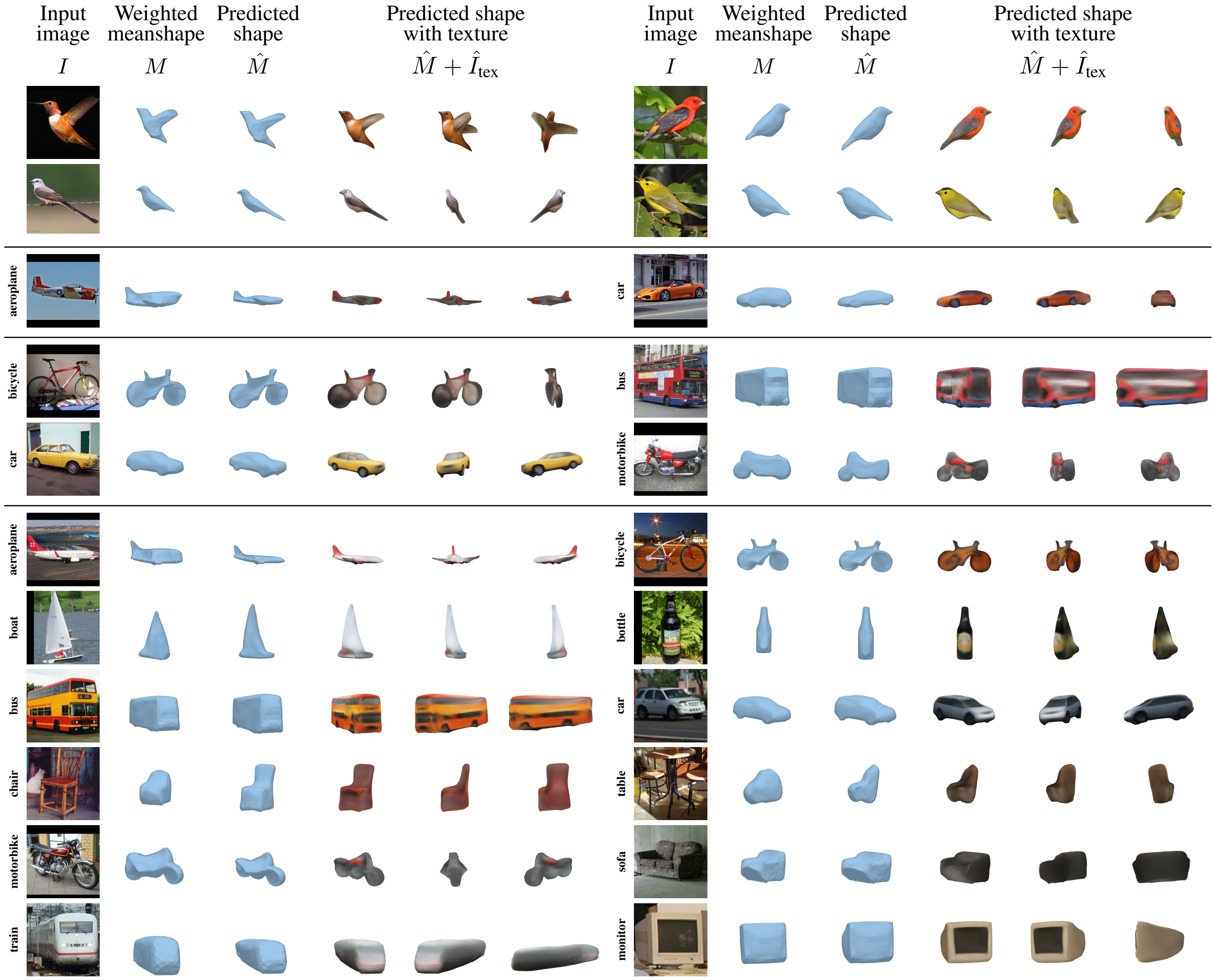

Figure 6: Qualitative results on different settings: CUB [49] (birds) and Pascal3D+ [54] (aeroplane and car, 4 automotive classes, all 12 classes). We show the input image I, the output M of the unsupervised shape selection module, the predicted shape $\hat{M}$ and the predicted textured shape $\hat{M} + \hat{I}_{\text{tex}}$ under several 3D rotations over the vertical axis of the predicted pose $\hat{\pi}$.

always lead to better scores, as in the case of fixed subdivision level 4. On the contrary, increasing the subdivision level during training leads to higher results in terms of both mask IoU and texture metrics. Indeed, dynamic subdivision allows to take advantage of low subdivision levels during the initial training phase – optimizing the shape smoothness in a faster and easier way – and at the same time leveraging the higher number of faces of high subdivision levels in the second part of the training – improving the finer details and the quality of the texture.

5. Conclusion

In this paper, we show how the 3D mesh reconstruction of objects can be learned jointly on multiple classes using only foreground masks and coarse camera poses as supervision. The proposed approach discerns between different object categories and learns meaningful category-level meanshapes, which were initialized as spheres, in an unsupervised manner. In addition, a novel approach to predict the instance-specific deformation at vertex level is presented. The network produces smooth deformations and is independent of the number of the mesh vertices, allowing the dynamic subdivision of the mesh during training. Quantitative and qualitative results on two public datasets show the effectiveness of the proposed method.

Acknowledgements. This research was supported by MIUR PRIN project “PREVUE: PRediction of activities and Events by Vision in an Urban Environment”, grant ID E94I19000650001.

References

[1] Tristan Aumentado-Armstrong, Alex Levinshtein, Stavros Tsogkas, Konstantinos G Derpanis, and Allan D Jepson. Cycle-consistent generative rendering for 2d-3d modality translation. In *International Conference on 3D Vision (3DV)*, pages 230–240, 2020. 2

[2] Wenzheng Chen, Huan Ling, Jun Gao, Edward Smith, Jaakko Lehtinen, Alec Jacobson, and Sanja Fidler. Learning to predict 3d objects with an interpolation-based differentiable renderer. *Advances in Neural Information Processing Systems*, 32:9609–9619, 2019. 2, 6

[3] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *European Conference on Computer Vision*. Springer, 2016. 1, 2

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 5

[5] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 2015. 5

[6] Rohit Girdhar, David F Fouhey, Mikel Rodriguez, and Abhinav Gupta. Learning a predictable and generative vector representation for objects. In *European Conference on Computer Vision*, pages 484–499. Springer, 2016. 1, 2

[7] Shubham Goel, Angjoo Kanazawa, and Jitendra Malik. Shape and viewpoint without keypoints. In *European Conference on Computer Vision*, 2020. 1, 2, 3, 4, 5, 6

[8] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. A papier-mâché approach to learning 3d surface generation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 216–224, 2018. 4

[9] JunYoung Gwak, Christopher B Choy, Manmohan Chandraker, Animesh Garg, and Silvio Savarese. Weakly supervised 3d reconstruction with adversarial constraint. In *International Conference on 3D Vision*. IEEE, 2017. 1, 2

[10] Bharath Hariharan, Pablo Arbeláez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *IEEE International Conference on Computer Vision*, pages 991–998, 2011. 5

[11] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *IEEE International Conference on Computer Vision*, pages 2961–2969, 2017. 5

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 4

[13] Paul Henderson and Vittorio Ferrari. Learning to generate and reconstruct 3d meshes with only 2d supervision. In *British Machine Vision Conference (BMVC)*, 2018. 2

[14] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *NeurIPS*, 2017. 6

[15] Eldar Insafutdinov and Alexey Dosovitskiy. Unsupervised learning of shape and pose with differentiable point clouds. In *Advances in Neural Information Processing Systems*, pages 2807–2817, 2018. 2

[16] Angjoo Kanazawa, Shubham Tulsiani, Alexei A Efros, and Jitendra Malik. Learning category-specific mesh reconstruction from image collections. In *European Conference on Computer Vision*, pages 371–386, 2018. 1, 2, 3, 5, 6

[17] Abhishek Kar, Shubham Tulsiani, Joao Carreira, and Jitendra Malik. Category-specific object reconstruction from a single image. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1966–1974, 2015. 2, 6

[18] Hiroharu Kato and Tatsuya Harada. Learning view priors for single-view 3d reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 2, 3

[19] Hiroharu Kato and Tatsuya Harada. Self-supervised learning of 3d objects from natural images. *preprint arXiv:1911.08850*, 2019. 2

[20] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3d mesh renderer. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3907–3916, 2018. 2

[21] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *preprint arXiv:1412.6980*, 2014. 5

[22] Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross Girshick. Pointrend: Image segmentation as rendering. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 9799–9808, 2020. 5, 7

[23] Nilesh Kulkarni, Abhinav Gupta, David F Fouhey, and Shubham Tulsiani. Articulation-aware canonical surface mapping. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 452–461, 2020. 2

[24] Nilesh Kulkarni, Abhinav Gupta, and Shubham Tulsiani. Canonical surface mapping via geometric cycle consistency. In *IEEE International Conference on Computer Vision*, pages 2202–2211, 2019. 2

[25] Jiahui Lei, Srinath Sridhar, Paul Guerrero, Minhyuk Sung, Niloy Mitra, and Leonidas J Guibas. Pix2surf: Learning parametric 3d surface models of objects from images. In *European Conference on Computer Vision*, 2020. 2

[26] Hai Li, Weicai Ye, Guofeng Zhang, Sanyuan Zhang, and Hujun Bao. Saliency guided subdivision for single-view mesh reconstruction. In *International Conference on 3D Vision (3DV)*, pages 1098–1107. IEEE, 2020. 2, 4

[27] Xueting Li, Sifei Liu, Shalini De Mello, Kihwan Kim, Xiaolong Wang, Ming-Hsuan Yang, and Jan Kautz. Online adaptation for consistent mesh reconstruction in the wild. In *NeurIPS*, 2020. 2

[28] Xueting Li, Sifei Liu, Kihwan Kim, Shalini De Mello, Varun Jampani, Ming-Hsuan Yang, and Jan Kautz. Self-supervised single-view 3d reconstruction via semantic consistency. In *European Conference on Computer Vision*, 2020. 2, 3, 5

[29] Chen-Hsuan Lin, Oliver Wang, Bryan C Russell, Eli Shechtman, Vladimir G Kim, Matthew Fisher, and Simon Lucey. Photometric mesh optimization for video-aligned 3d object reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 969–978, 2019. 1, 2

[30] Shichen Liu, Tianye Li, Weikai Chen, and Hao Li. Soft rasterizer: A differentiable renderer for image-based 3d reasoning. In *IEEE International Conference on Computer Vision*, pages 7708–7717, 2019. 2, 3

[31] Matthew M Loper and Michael J Black. Opendr: An approximate differentiable renderer. In *European Conference on Computer Vision*, pages 154–169. Springer, 2014. 2

[32] Mario Lucic, Karol Kurach, Marcin Michalski, Sylvain Gelly, and Olivier Bousquet. Are gans created equal? a large-scale study. In *NeurIPS*, 2018. 6

[33] Priyanka Mandikal, Navaneet KL, and R Venkatesh Babu. 3d-psrnet: Part segmented 3d point cloud reconstruction from a single image. In *European Conference on Computer Vision (ECCV) Workshops*, 2018. 2

[34] Andrea Palazzi, Luca Bergamini, Simone Calderara, and Rita Cucchiara. End-to-end 6-dof object pose estimation through differentiable rasterization. In *European Conference on Computer Vision (ECCV) Workshops*, 2018. 2

[35] Andrea Palazzi, Luca Bergamini, Simone Calderara, and Rita Cucchiara. Warp and learn: Novel views generation for vehicles and other objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 5

[36] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2019. 4

[37] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2337–2346, 2019. 5

[38] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *Neural Information Processing Systems Workshops*, 2017. 5

[39] Dario Pavllo, David Grangier, and Michael Auli. Quaternet: A quaternion-based recurrent model for human motion. In *British Machine Vision Conference, BMVC 2018*, 2018. 4

[40] Stephan R Richter and Stefan Roth. Matryoshka networks: Predicting 3d geometry via nested shape layers. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1936–1944, 2018. 2

[41] Alessandro Simoni, Luca Bergamini, Andrea Palazzi, Simone Calderara, and Rita Cucchiara. Future urban scenes generation through vehicles synthesis. In *2020 25th International Conference on Pattern Recognition*. IEEE, 2021. 5

[42] Alessandro Simoni, Andrea D'Eusanio, Stefano Pini, Guido Borghi, and Roberto Vezzani. Improving car model classification through vehicle keypoint localization. In *International Conference on Computer Vision Theory and Applications*, 2021. 5

[43] Olga Sorkine. Differential representations for mesh processing. In *Computer Graphics Forum*, 2006. 4

[44] Xingyuan Sun, Jiajun Wu, Xiuming Zhang, Zhoutong Zhang, Chengkai Zhang, Tianfan Xue, Joshua B Tenenbaum, and William T Freeman. Pix3d: Dataset and methods for single-image 3d shape modeling. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 5

[45] Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. Multi-view 3d models from single images with a convolutional network. In *European Conference on Computer Vision*, pages 322–337, 2016. 1, 2

[46] Shubham Tulsiani, Alexei A Efros, and Jitendra Malik. Multi-view consistency as supervisory signal for learning shape and pose prediction. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2897–2905, 2018. 2

[47] Shubham Tulsiani, Nilesh Kulkarni, and Abhinav Gupta. Implicit mesh reconstruction from unannotated image collections. *preprint arXiv:2007.08504*, 2020. 1, 2, 5, 6

[48] Shubham Tulsiani, Tinghui Zhou, Alexei A Efros, and Jitendra Malik. Multi-view supervision for single-view reconstruction via differentiable ray consistency. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2626–2634, 2017. 1, 2, 5, 6

[49] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 2, 5, 6, 8

[50] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *European Conference on Computer Vision*, pages 52–67, 2018. 1, 2, 4

[51] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 6

[52] Olivia Wiles and Andrew Zisserman. Silnet : Single- and multi-view reconstruction by learning from silhouettes. In *British Machine Vision Conference (BMVC)*, 2017. 2

[53] Shangzhe Wu, Christian Rupprecht, and Andrea Vedaldi. Unsupervised learning of probably symmetric deformable 3d objects from images in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 5

[54] Yu Xiang, Roozbeh Mottaghi, and Silvio Savarese. Beyond pascal: A benchmark for 3d object detection in the wild. In *IEEE Winter Conference on Applications of Computer Vision*, pages 75–82, 2014. 2, 5, 6, 7, 8

[55] Haozhe Xie, Hongxun Yao, Xiaoshuai Sun, Shangchen Zhou, and Shengping Zhang. Pix2vox: Context-aware 3d reconstruction from single and multi-view images. In *IEEE International Conference on Computer Vision*, 2019. 1, 2

[56] Xinchen Yan, Jimei Yang, Ersin Yumer, Yijie Guo, and Honglak Lee. Perspective transformer nets: learning single-view 3d object reconstruction without 3d supervision. In *NeurIPS*, 2016. 2

[57] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 5

[58] Rui Zhu, Hamed Kiani Galoogahi, Chaoyang Wang, and Simon Lucey. Rethinking reprojection: Closing the loop for pose-aware shape reconstruction from a single image. In *IEEE International Conference on Computer Vision*, 2017. 2