Learning Category-Specific Mesh Reconstruction from Image Collections

Angjoo Kanazawa*, Shubham Tulsiani*, Alexei A. Efros, Jitendra Malik

University of California, Berkeley
{kanazawa,shubhtuls,efros,malik}@eecs.berkeley.edu

Abstract. We present a learning framework for recovering the 3D shape, camera, and texture of an object from a single image. The shape is represented as a deformable 3D mesh model of an object category where a shape is parameterized by a learned mean shape and per-instance predicted deformation. Our approach allows leveraging an annotated image collection for training, where the deformable model and the 3D prediction mechanism are learned without relying on ground-truth 3D or multi-view supervision. Our representation enables us to go beyond existing 3D prediction approaches by incorporating texture inference as prediction of an image in a canonical appearance space. Additionally, we show that semantic keypoints can be easily associated with the predicted shapes. We present qualitative and quantitative results of our approach on CUB and PASCAL3D datasets and show that we can learn to predict diverse shapes and textures across objects using only annotated image collections. The project website can be found at https://akanazawa.github.io/cmr/.

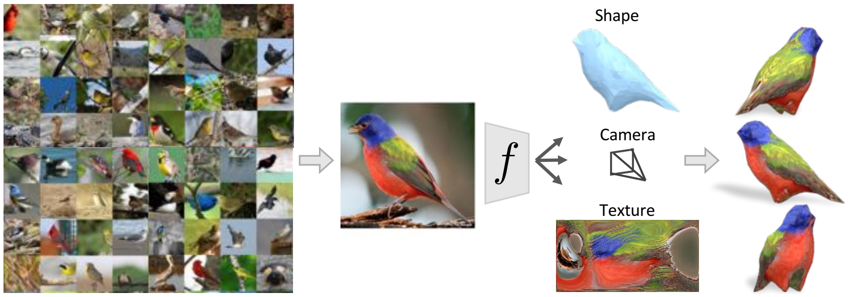

Fig. 1: Given an annotated image collection of an object category, we learn a predictor f that can map a novel image I to its 3D shape, camera pose, and texture.

1 Introduction

Consider the image of the bird in Figure 1. Even though this flat two-dimensional picture printed on a page may be the first time we are seeing this particular bird, we can

[*] The first two authors procrastinated equally on this work.

infer its rough 3D shape, understand the camera pose, and even guess what it would look like from another view. We can do this because all the previously seen birds have enabled us to develop a mental model of what birds are like, and this knowledge helps us to recover the 3D structure of this novel instance.

In this work, we present a computational model that can similarly learn to infer a 3D representation given just a single image. As illustrated in Figure 1, the learning only relies on an annotated 2D image collection of a given object category, comprising of foreground masks and semantic keypoint labels. Our training procedure, depicted in Figure 2, forces a common prediction model to explain all the image evidences across many examples of an object category. This allows us to learn a meaningful 3D structure despite only using a single-view per training instance, without relying on any ground-truth 3D data for learning.

At inference, given a single unannotated image of a novel instance, our learned model allows us to infer the shape, camera pose, and texture of the underlying object. We represent the shape as a 3D mesh in a canonical frame, where the predicted camera transforms the mesh from this canonical space to the image coordinates. The particular shape of each instance is instantiated by deforming a learned category-specific mean shape with instance-specific predicted deformations. The use of this shared 3D space affords numerous advantages as it implicitly enforces correspondences across 3D representations of different instances. As we detail in Section 2, this allows us to formulate the task of inferring mesh texture of different objects as that of predicting pixel values in a common texture representation. Furthermore, we can also easily associate semantic keypoints with the predicted 3D shapes.

Our shape representation is an instantiation of deformable models, the history of which can be traced back to D'Arcy Thompson [29], who in turn was inspired by the work of Dürer [6]. Thompson observed that shapes of objects of the same category may be aligned through geometrical transformations. Cootes and Taylor [5] operationalized this idea to learn a class-specific model of deformation for 2D images. Pioneering work of Blanz and Vetter [2] extended these ideas to 3D shapes to model the space of faces. These techniques have since been applied to model human bodies [1,19], hands [27,17], and more recently on quadruped animals [40]. Unfortunately, all of these approaches require a large collection of 3D data to learn the model, preventing their application to categories where such data collection is impractical. In contrast, our approach is able to learn using only an annotated image collection.

Sharing our motivation for relaxing the requirement of 3D data to learn morphable models, some related approaches have examined the use of similarly annotated image collections. Cashman and Fitzgibbon [3] use keypoint correspondences and segmentation masks to learn a morphable model of dolphins from images. Kar *et al.* [15] extend this approach to general rigid object categories. Both approaches follow a *fitting-based* inference procedure, which relies on mask (and optionally keypoint) annotations at test-time and is computationally inefficient. We instead follow a *prediction-based* inference approach, and learn a parametrized predictor which can directly infer the 3D structure from an unannotated image. Moreover, unlike these approaches, we also address the task of texture prediction which cannot be easily incorporated with these methods.

While deformable models have been a common representation for 3D inference, the recent advent of deep learning based prediction approaches has resulted in a plethora of alternate representations being explored using varying forms of supervision. Relying on ground-truth 3D supervision (using synthetic data), some approaches have examined learning voxel [4,8,39,33], point cloud [7] or octree [10,26] prediction. While some learning based methods do pursue mesh prediction [14,35,18,24], they also rely on 3D supervision which is only available for restricted classes or in a synthetic setting. Reducing the supervision to multi-view masks [34,21,30,9] or depth images [30] has been explored for voxel prediction, but the requirement of multiple views per instance is still restrictive. While these approaches show promising results, they rely on stronger supervision (ground-truth 3D or multi-view) compared to our approach.

In the context of these previous approaches, the proposed approach differs primarily in three aspects:

- *Shape representation and inference method.* We combine the benefits of the classically used deformable mesh representations with those of a learning based prediction mechanism. The use of a deformable mesh based representation affords several advantages such as memory efficiency, surface-level reasoning and correspondence association. Using a learned prediction model allows efficient inference from a single unannotated image
- *Learning from an image collection.* Unlike recent CNN based 3D prediction methods which require either ground-truth 3D or multi-view supervision, we only rely on an annotated image collection, with only one available view per training instance, to learn our prediction model.
- *Ability to infer texture.* There is little past work on predicting the 3D shape and the texture of objects from a single image. Recent *prediction-based* learning methods use representations that are not amenable to textures (*e.g.* voxels). The classical deformable model *fitting-based* approaches cannot easily incorporate texture for generic objects. An exception is texture inference on human faces [2,22,23,28], but these approaches require a large-set of 3D ground truth data with high quality texture maps. Our approach enables us to pursue the task of texture inference from image collections alone, and we address the related technical challenges regarding its incorporation in a learning framework.

2 Approach

We aim to learn a predictor f_θ (parameterized as a CNN) that can infer the 3D structure of the underlying object instance from a single image I. The prediction $f_\theta(I)$ is comprised of the 3D shape of the object in a canonical frame, the associated texture, as well as the camera pose. The shape representation we pursue in this work is of the form of a 3D mesh. This representation affords several advantages over alternates like probabilistic volumetric grids *e.g.* amenability to texturing, correspondence inference, surface level reasoning and interpretability.

The overview of the proposed framework is illustrated in Figure 2. The input image is passed through an encoder to a latent representation that is shared by three modules

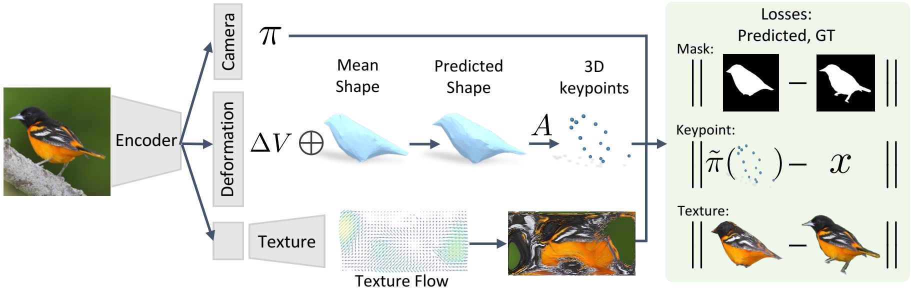

Fig. 2: **Overview of the proposed framework.** An image I is passed through a convolutional encoder to a latent representation that is shared by modules that estimate the camera pose, deformation and texture parameters. Deformation is an offset to the learned mean shape, which when added yield instance specific shapes in a canonical coordinate frame. We also learn correspondences between the mesh vertices and the semantic keypoints. Texture is parameterized as an UV image, which we predict through texture flow (see Section 2.3). The objective is to minimize the distance between the rendered mask, keypoints and textured rendering with the corresponding ground truth annotations. We do not require ground truth 3D shapes or multi-view cues for training.

that estimate the camera pose, shape deformation, and texture parameters. The deformation is added to the learned category-level mean shape to obtain the final predicted shape. The objective of the network is to minimize the corresponding losses when the shape is rendered onto the image. We train a separate model for each object category.

We first present the representations predicted by our model in Section 2.1, and then describe the learning procedure in Section 2.2. We initially present our framework for predicting shape and camera pose, and then describe how the model is extended to predict the associated texture in Section 2.3.

2.1 Inferred 3D Representation

Given an image I of an instance, we predict $f_\theta(I) \equiv (M, \pi)$, a mesh M and camera pose π to capture the 3D structure of the underlying object. In addition to these directly predicted aspects, we also learn the association between the mesh vertices and the category-level semantic keypoints. We describe the details of the inferred representations below.

Shape Parametrization. We represent the shape as a 3D mesh $M \equiv (V, F)$, defined by vertices $V \in \mathbb{R}^{|V| \times 3}$ and faces F. We assume a fixed and pre-determined mesh connectivity, and use the faces F corresponding to a spherical mesh. The vertex positions V are instantiated using (learned) instance-independent mean vertex locations $\bar{V}$ and instance-dependent predicted deformations Δ_V, which when added, yield instance vertex locations $V = \bar{V} + \Delta_V$. Intuitively, the mean shape $\bar{V}$ can be considered as a learnt bias term for the predicted shape V.

Camera Projection. We model the camera with weak-perspective projection and predict, from the input image I, the scale $s \in \mathbb{R}$, translation $\mathbf{t} \in \mathbb{R}^2$, and rotation (captured by quaternion $\mathbf{q} \in \mathbb{R}^4$). We use $\pi(P)$ to denote the projection of a set of 3D points P onto the image coordinates via the weak-perspective projection defined by $\pi \equiv (s, \mathbf{t}, \mathbf{q})$.

Associating Semantic Correspondences. As we represent the shape using a category-specific mesh in the canonical frame, the regularities across instances encourage semantically consistent vertex positions across instances, thereby implicitly endowing semantics to these vertices. We can use this insight and learn to explicitly associate semantic keypoints *e.g.*, beak, legs *etc.* with the mesh via a keypoint assignment matrix $A \in \mathcal{R}_{+}{}^{|K| \times |V|}$ s.t. $\sum_v A_{k,v} = 1$. Here, each row A_k represents a probability distribution over the mesh vertices of corresponding to keypoint k, and can be understood as approximating a one-hot vector of vertex selection for each keypoint. As we describe later in our learning formulation, we encourage each A_k to be a peaked distribution. Given the vertex positions V, we can infer the location v_k for the k^{th} keypoint as $v_k = \sum_v A_{k,v} v$. More concisely, the keypoint locations induced by vertices V can be obtained as $A \cdot V$. We initialize the keypoint assignment matrix A uniformly, but over the course of training it learns to better associate semantic keypoints with appropriate mesh vertices.

In summary, given an image I of an instance, we predict the corresponding camera π and the shape deformation Δ_V as $(\pi, \Delta_V) = f(I)$. In addition, *we also learn* (across the dataset), instance-independent parameters $\{\bar{V}, A\}$. As described above, these category-level (learned) parameters, in conjunction with the instances-specific predictions, allow us to recover the mesh vertex locations V and coordinates of semantic keypoints $A \cdot V$.

2.2 Learning from an Image Collection

We present an approach to train f_θ without relying on strong supervision in the form of ground truth 3D shapes or multi-view images of an object instance. Instead, we guide the learning from an image collection annotated with sparse keypoints and segmentation masks. Such a setting is more natural and easily obtained, particularly for animate and deformable objects such as birds or animals. It is extremely difficult to obtain scans, or even multiple views of the same instance for these classes, but relatively easier to acquire a single image for numerous instances.

Given the annotated image collection, we train f_θ by formulating an objective function that consists of instance specific losses and priors. The instance-specific energy terms ensure that the predicted 3D structure is consistent with the available evidence (masks and keypoints) and the priors encourage generic desired properties *e.g.* smoothness. As we learn a common prediction model f_θ across many instances, the common structure across the category allows us to learn meaningful 3D prediction despite only having a single-view per instance.

Training Data. We assume an annotated training set $\{(I_i, S_i, x_i)\}_{i=1}^{N}$ for each object category, where I_i is the image, S_i is the instance segmentation, and $x_i \in \mathbb{R}^{2 \times K}$ is the set of K keypoint locations. As previously leveraged by [31,15], applying structure-from-motion to the annotated keypoint locations additionally allows us to obtain a rough estimate of the weak-perspective camera $\tilde{\pi}_i$ for each training instance. This results in an augmented training set $\{(I_i, S_i, x_i, \tilde{\pi}_i)\}_{i=1}^{N}$, which we use for training our predictor f_θ.

Instance Specific Losses. We ensure that the predicted 3D structure matches the available annotations. Using the semantic correspondences associated to the mesh via the keypoint assignment matrix A, we formulate a keypoint reprojection loss. This term encourages the predicted 3D keypoints to match the annotated 2D keypoints when projected onto the image:

$$L_{\texttt{reproj}} = \sum_i ||x_i - \tilde{\pi}_i(AV_i)||_2. \tag{1}$$

Similarly, we enforce that the predicted 3D mesh, when rendered in the image coordinates, is consistent with the annotated foreground mask: $L_{\texttt{mask}} = \sum_i ||S_i - \mathcal{R}(V_i, F, \tilde{\pi}_i)||_2$. Here, $\mathcal{R}(V, F, \pi)$ denotes a rendering of the segmentation mask image corresponding to the 3D mesh $M = (V, F)$ when rendered through camera π. In all of our experiments, we use Neural Mesh Renderer [16] to provide a differentiable implementation of $\mathcal{R}(\cdot)$.

We also train the predicted camera pose to match the corresponding estimate obtained via structure-from-motion using a regression loss $L_{\texttt{cam}} = \sum_i ||\tilde{\pi}_i - \pi_i||_2$. We found it advantageous to use the structure-from-motion camera $\tilde{\pi}_i$, and not the predicted camera π_i, to define $L_{\texttt{mask}}$ and $L_{\texttt{reproj}}$ losses. This is because during training, in particular the initial stages when the predictions are often incorrect, an error in the predicted camera can lead to high errors despite accurate shape, and possibly adversely affect learning.

Priors. In addition to the data-dependent losses which ensure that the predictions match the evidence, we leverage generic priors to encourage additional properties. The prior terms that we use are:
Smoothness. In the natural world, shapes tend to have a smooth surface and we would like our recovered 3D shapes to behave similarly. An advantage of using a mesh representation is that it naturally affords reasoning at the surface level. In particular, enforcing smooth surface has been extensively studied by the Computer Graphics community [20,25]. Following the literature, we formulate surface smoothness as minimization of the mean curvature. On meshes, this is captured by the norm of the graph Laplacian, and can be concisely written as $L_{\texttt{smooth}} = ||LV||_2$, where L is the discrete Laplace-Beltrami operator. We construct L once using the connectivity of the mesh and this can be expressed as a simple linear operator on vertex locations. See appendix for details.
Deformation Regularization. In keeping with a common practice across deformable model approaches [2,3,15], we find it beneficial to regularize the deformations as it discourages arbitrarily large deformations and helps learn a meaningful mean shape. The corresponding energy term is expressed as $L_{\texttt{def}} = ||\Delta_V||_2$.
Keypoint association. As discussed in Section 2.1, we encourage the keypoint assignment matrix A to be a peaked distribution as it should intuitively correspond to a one-hot vector. We therefore minimize the average entropy over all keypoints: $L_{\texttt{vert2kp}} = \frac{1}{|K|}\sum_k \sum_v -A_{k,v} \log A_{k,v}$.

In summary, the overall objective for shape and camera is

$$L = L_{\texttt{reproj}} + L_{\texttt{mask}} + L_{\texttt{cam}} + L_{\texttt{smooth}} + L_{\texttt{def}} + L_{\texttt{vert2kp}}. \tag{2}$$

Symmetry Constraints. Almost all common object categories, including the ones we consider, exhibit reflectional symmetry. To exploit this structure, we constrain the predicted shape and deformations to be mirror-symmetric. As our mesh topology corresponds to that of a sphere, we identify symmetric vertex pairs in the initial topology. Given these pairs, we only learn/predict parameters for one vertex in each pair for the mean shape $\bar{V}$ and deformations Δ_V. See appendix for details.

Initialization and Implementation Details. While our mesh topology corresponds to a sphere, following previous fitting based deformable model approaches [15], we observe that a better initialization of the mean vertex positions $\bar{V}$ speeds up learning. We compute the convex hull of the mean keypoint locations obtained during structure-from-motion and initialize the mean vertex locations to lie on this convex hull – the procedure is described in more detail in the appendix. As the different energy terms in Eq. 2 have naturally different magnitudes, we weight them accordingly to normalize their contribution.

2.3 Incorporating Texture Prediction

In our formulation, all recovered shapes share a common underlying 3D mesh structure – each shape is a deformation of the mean shape. We can leverage this property to reduce texturing of a particular instance to predicting the texture of the mean shape. Our mean shape is isomorphic to a sphere, whose texture can be represented as an image I^{uv}, the values of which get mapped onto the surface via a fixed UV mapping (akin to unrolling a globe into a flat map) [13]. Therefore, we formulate the task of texture prediction as that of inferring the pixel values of I^{uv}. This image can be thought of as a canonical appearance space of the object category. For example, a particular triangle on the predicted shape always maps to a particular region in I^{uv}, irrespective of how it was deformed. This is illustrated in Figure 3. In this texture parameterization, each pixel in the UV image has a consistent semantic meaning, thereby making it easier for the prediction model to leverage common patterns such as correlation between the bird back and the body color.

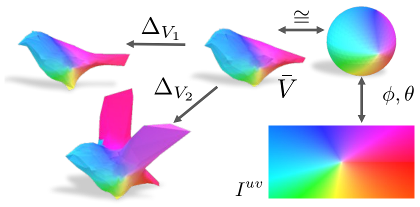

Fig. 3: **Illustration of the UV mapping.** We illustrate how a texture image I^{uv} can induce a corresponding texture on the predicted meshes. A point on a sphere can be mapped onto the image I^{uv} via using spherical coordinates. As our mean shape has the same mesh geometry (vertex connectivity) as a sphere we can transfer this mapping onto the mean shape. The different predicted shapes, in turn, are simply deformations of the mean shape and can use the same mapping.

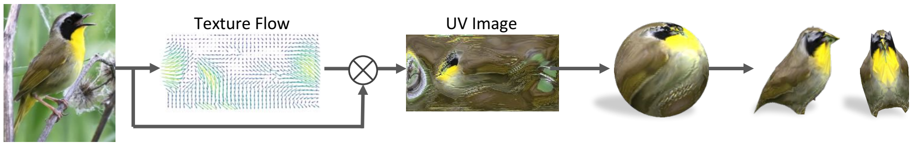

Fig. 4: **Illustration of texture flow.** We predict a texture flow $\mathcal{F}$ that is used to bilinearly sample the input image I to generate the texture image I^{uv}. We can use this predicted UV image I^{uv} to then texture the instance mesh via the UV mapping procedure illustrated in Figure 3.

We incorporate texture prediction module into our framework by setting up a decoder that upconvolves the latent representation to the spatial dimension of I^{uv}. While directly regressing the pixel values of I^{uv} is a feasible approach, this often results in blurry images. Instead, we take inspiration from [38] and formulate this task as that of predicting the appearance flow. Instead of regressing the pixel values of I^{uv}, the texture module outputs where to copy the color of the pixel from the original input image. This prediction mechanism, depicted in Figure 4, easily allows our predicted texture to retain the details present in the input image. We refer to this output as 'texture flow' $\mathcal{F} \in \mathbb{R}^{H_{uv} \times W_{uv} \times 2}$, where H_{uv}, W_{uv} are the height and width of I^{uv}, and $\mathcal{F}(u, v)$ indicates the (x, y) coordinates of the input image to sample the pixel value from. This allows us to generate the UV image $I^{uv} = G(I; \mathcal{F})$ by bilinear sampling G of the original input image I according to the predicted flow $\mathcal{F}$. This is illustrated in Figure 4.

Now we formulate our texture loss, which encourages the rendered texture image to match the foreground image:

$$L_{\texttt{texture}} = \sum_i \text{dist}(S_i \odot I_i, S_i \odot \mathcal{R}(V_i, F, \tilde{\pi}_i, I^{uv})). \tag{3}$$

$\mathcal{R}(V_i, F, \tilde{\pi}_i, I_i^{uv})$ is the rendering of the 3D mesh with texture defined by I^{uv}. We use the perceptual metric of Zhang *et al.* [37] as the distance metric.

The loss function above provides supervisory signals to regions of I^{uv} corresponding to the foreground portion of the image, but not to other regions of I^{uv} corresponding to parts that are not directly visible in the image. While the common patterns across the dataset *e.g.* similar colors for bird body and back can still allow meaningful prediction, we find it helpful to add a further loss that encourages the texture flow to select pixels only from the foreground region in the image. This can be simply expressed by sampling the distance transform field of the foreground mask $\mathcal{D}_S$ (where for all points x in the foreground, $\mathcal{D}_S(x) = 0$) according to $\mathcal{F}$ and summing the resulting image:

$$L_{\texttt{dt}} = \sum_i \sum_{u,v} G(\mathcal{D}_{S_i}; \mathcal{F}_i)(u, v). \tag{4}$$

In contrast to inferring the full texture map, directly sampling the actual pixel values that the predicted mesh projects onto creates holes and leaking of the background texture at the boundaries. Similarly to the shape parametrization, we also explicitly encode symmetry in our I^{uv} prediction, where symmetric faces gets mapped on to the same UV coordinate in I^{uv}. Additionally, we only back-propagate gradients from $L_{\texttt{texture}}$ to the predicted texture (and not the predicted shape) since bilinear sampling often results in high-frequency gradients that destabilize shape learning. Our shape prediction is therefore learned only using the objective in Eq. 2, and the losses $L_{\texttt{texture}}$ and $L_{\texttt{dt}}$ can be viewed as encouraging prediction of correct texture 'on top' of the learned shape.

3 Experiments

We demonstrate the ability of our presented approach to learn single-view inference of shape, texture and camera pose using only a category-level annotated image collection. As a running example, we consider the 'bird' object category as it represents a challenging scenario that has not been addressed via previous approaches. We first present, in Section 3.1, our experimental setup, describing the annotated image collection and CNN architecture used.

As ground-truth 3D is not available for benchmarking, we present extensive qualitative results in Section 3.2, demonstrating that we learn to predict meaningful shapes and textures across birds. We also show we capture the shape deformation space of the category and that the implicit correspondences in the deformable model allow us to have applications like texture transfer across instances.

We also present some quantitative results to provide evidence for the accuracy of our shape and camera estimates in Section 3.3. While there has been little work for reconstructing categories like birds, some approaches have examined the task of learning shape prediction using an annotated image collection for some rigid classes. In Section 3.4 we present our method's results on some additional representative categories, and show that our method performs comparably, if not better than the previously proposed alternates while having several additional advantages *e.g.* learning semantic keypoints and texture prediction.

3.1 Experimental Setup

Dataset. We use the CUB-200-2011 dataset [32], which has 6000 training and test images of 200 species of birds. Each image is annotated with the bounding box, visibility indicator and locations of 14 semantic keypoints, and the ground truth foreground mask. We filter out nearly 300 images where the visible number of keypoints are less than or equal to 6, since these typically correspond to truncated close shots. We divide the test set in half to create a validation set, which we use for hyper-parameter tuning.

Network Architecture. A schematic of the various modules of our prediction network is depicted in Figure 2. The encoder consists of an ImageNet pretrained ResNet-18 [12], followed by a convolutional layer that downsamples the spatial and the channel dimensions by half. This is vectorized to form a 4096-D vector, which is sent to two fully-connected layers to get to the shared latent space of size 200. The deformation and the camera prediction components are linear layers on top of this latent space. The texture flow component consists of 5 upconvolution layers where the final output is passed through a $tanh$ function to keep the flow in a normalized [-1, 1] space. We use the neural mesh renderer [16] so all rendering procedures are differentiable. All images

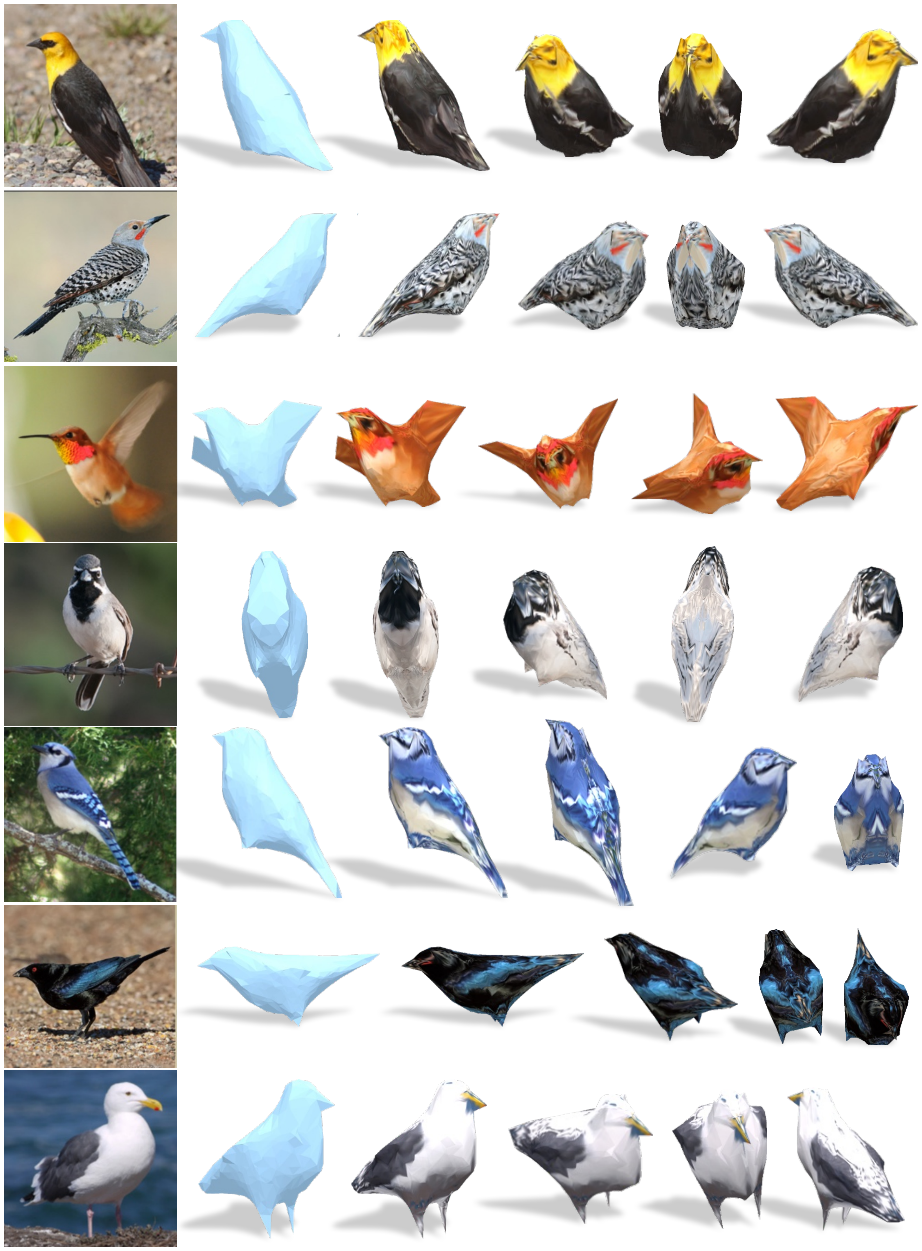

Fig. 5: **Sample results.** We show predictions of our approach on images from the test set. For each input image on the left, we visualize (in order): the predicted 3D shape and texture viewed from the predicted camera, and textured shape from three novel viewpoints. See the appendix for additional randomly selected results and video at https://akanazawa.github.io/cmr/.

are cropped using the instance bounding box and resized such that the maximum image dimension is 256. We augment the training data on the fly by jittering the scale and translation of the bounding box and with image mirroring. Our mesh geometry corresponds to that of a perfectly symmetric sphere with 642 vertices and 1280 faces.

3.2 Qualitative Results

We visualize the results and application of our learned predictor using the CUB dataset. We show various reconstructions corresponding to different input images, visualize some of the deformation modes learned, and show that the common deformable model parametrization allows us to transfer the texture of one instance onto another.

Single-view 3D Reconstruction. We show sample reconstruction results on images from the CUB test set in Figure 5. We show the predicted shape and texture from the inferred camera viewpoint, as well as from novel views. Please see appendix for additional randomly selected samples and videos showing the results from 360 views.

We observe that our learned model can accurately predict the shape, estimate the camera and also infer meaningful texture from the corresponding input image. Our predicted 3D shape captures the overall shape (fat or thin birds), and even some finer details *e.g.* beaks or large deformations *e.g.* flying birds. Additionally, our learned pose and texture prediction are accurate and realistic across different instances. We observe that the error modes corresponds to not predicting rare poses, and inability to incorporate asymmetric articulation. However, we feel that these predictions learned using only an annotated image collection are encouraging.

Learned shape space. The presented approach represents the shape of an instance via a category-level learned mean shape and a per-instance predicted deformation Δ_V. To gain insight into the common modes of deformation captured via our predictor, obtained the principal deformation modes by computing PCA on the predicted deformations across all instances in the training set.

We visualize in Figure 6 our mean shape deformed in directions corresponding three common deformation modes. We note that these plausibly correspond to some of the natural factors of variation in the 3D structure across birds *e.g.* fat or thin birds, opening of wings, deformation of tails and legs.

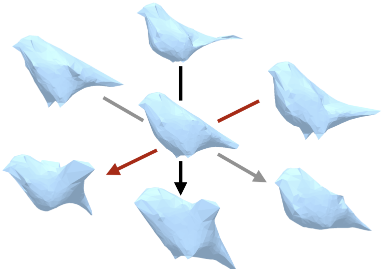

Fig. 6: **Learned deformation modes.** We visualize the space of learned shapes by depicting the mean shape (centre) and three common modes of deformation as obtained by PCA on the predicted deformations across the dataset.

Texture Transfer. Recall that the textures of different instance in our formulation are captured in a canonical appearance space in the form of a predicted ‘texture image’

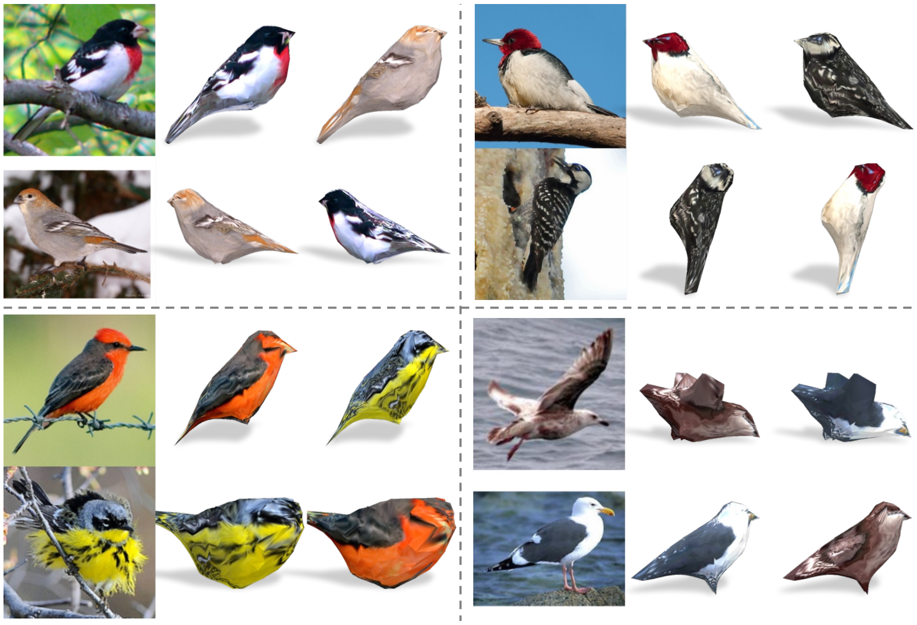

Fig. 7: **Texture Transfer Results.** Our representation allows us to easily transfer the predicted texture across instances using the canonical appearance image (see text for details). We visualize sample results of texture transfer across different pairs of birds. For each pair, we show (left): the input image, (middle): the predicted textured mesh from the predicted viewpoint, and (right): the predicted mesh textured using the predicted texture of the other bird.

I_{uv}. This parametrization allows us to easily modify the surface appearance, and in particular transfer texture across instances.

We show some results in Figure 7 where we sample pairs of instances, and transfer the texture from one image onto the predicted shape of the other. We can achieve this by simply using the predicted texture image corresponding to the first when rendering the predicted 3D for the other. We note that even though the two views might be different, since the underlying 'texture image' space is consistent, the transferred texture is also semantically consistent *e.g.* the colors corresponding to the one bird's body are transferred onto the other bird's body.

3.3 Quantitative Evaluation

We attempt to indirectly measure the quality of our recovered reconstructions on the CUB dataset. As there is no ground-truth 3D available for benchmarking, we instead evaluate the mask reprojection accuracy. For each test instance in the CUB dataset, we obtain a mask prediction via rendering the predicted 3D shape from the predicted camera viewpoint. We then compute the intersection over union (IoU) of this predicted mask with the annotated ground-truth mask. Note that to correctly predict the foreground mask, we need both, accurate shape and accurate camera.

Our results are plotted in Figure 8. We compare the accuracy our full shape prediction (using learned mean shape $\bar{V}$ and predicted deformation Δ_V) against only using the learned mean shape to obtain the predicted mask. We observe that the predicted deformations result in improvements, indicating that we are able to capture the specifics of the shape of different instances. Additionally, we also report the performance using the camera obtained via structure from motion (which uses ground-truth annotated keypoints) instead of using the predicted camera. We note that comparable results in the two settings demonstrate the accuracy of our learned camera estimation. Lastly, we can also measure our keypoint reprojection accuracy using the percentage of correct keypoints (PCK) metric [36]. We similarly observe that our full predicted shape performs (slightly) better than only relying on the category-level mean shape – by obtaining a PCK (at normalized distance threshold 0.1) of 0.81 compared to 0.80. The improvement over the mean shape is less prominent in this scenario as most of the semantic keypoints defined are on the torso and therefore typically undergo only small deformations.

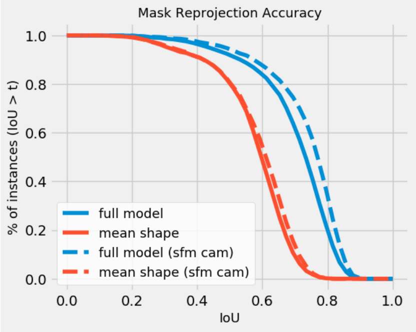

Fig. 8: **Mask reprojection accuracy evaluation on CUB.** We plot the fraction of test instances with IoU between the predicted and ground-truth mask higher than different thresholds (higher is better) and compare the predictions using the full model against only using the learned mean shape. We report the reprojection accuracy using predicted cameras and cameras obtained via structure-from-motion based on keypoint annotation.

Method	Aeroplane	Car
CSDM [15]	0.40	0.60
DRC [30]	0.42	0.67
Ours	0.46	0.64

Table 1: **Reconstruction evaluation using PASCAL 3D+.** We report the mean intersection over union (IoU) on PASCAL 3D+ to benchmark the obtained 3D reconstructions (higher is better). We compare to previous deformable model fitting-based [15] and volumetric prediction [30] approaches that use similar image collection supervision. Note that our approach can additionally predict texture and semantics.

3.4 Evaluation on Other Object Classes

While our primary results focus on predicting the 3D shape and texture of birds using the CUB dataset, we note that some previous approaches have examined the task of shape inference/prediction using a similar annotated image collection as supervision. While these previous methods do not infer texture, we can compare our shape predictions against those obtained by these techniques.

We compare to previous deformable model fitting-based [15] and volumetric prediction [30] methods using the PASCAL 3D+ dataset and examine the car and aeroplane

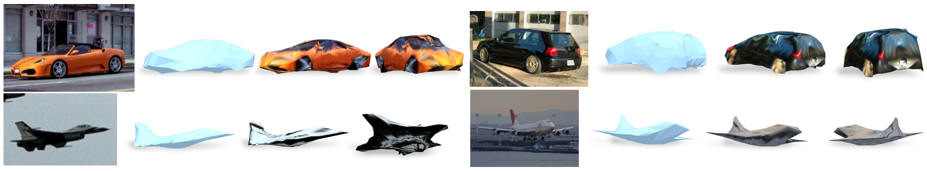

Fig. 9: **Pascal 3D+ results.** We show predictions of our approach on images from the test set. For each input image on the left, we visualize (in order): the predicted 3D shape viewed from the predicted camera, the predicted shape with texture viewed from the predicted camera, and the shape with texture viewed from a novel viewpoint.

categories. Both of these approaches can leverage the annotation we have available *i.e.* segmentation masks and keypoints to learn 3D shape inference (although [30] requires annotated cameras instead of keypoints). Similar to [30], we use PASCAL VOC and Imagenet images with available keypoint annotations from PASCAL3D+ to train our model, and use an off-the shelf segmentation algorithm [11] to obtain foreground masks for the ImageNet subset.

We report the mean IoU evaluation on the test set in Table 1 and observe that we perform comparably, if not better than these alternate methods. We also note that our approach yields additional outputs *e.g.* texture, that these methods do not. We visualize some predictions in Figure 9. While our predicted shapes are often reasonable, the textures have more errors due to shiny regions (*e.g.* for cars) or smaller amount of training data (*e.g.* for aeroplanes).

4 Discussion

We have presented a framework for learning single-view prediction of a textured 3D mesh using an image collection as supervision. While our results represent an encouraging step, we have by no means solved the problem in the general case, and a number of interesting challenges and possible directions remain. Our formulation addresses shape change and articulation via a similar shape deformation mechanism, and it may be beneficial to extend our deformable shape model to explicitly allow articulation. Additionally, while we presented a method to synthesize texture via copying image pixels, a more sophisticated mechanism that allows both, copying image content and synthesizing novel aspects might be desirable. Finally, even though we can learn using only a single-view per training instance, our approach may be equally applicable, and might yield perhaps even better results, for the scenario where multiple views per training instance are available. However, on the other end of the supervision spectrum, it would be desirable to relax the need of annotation even further, and investigate learning similar prediction models using unannotated image collections.

Acknowledgements. We thank David Fouhey for the creative title suggestions and members of the BAIR community for helpful discussions and comments. This work was supported in part by Intel/NSF VEC award IIS-1539099, NSF Award IIS-1212798, and BAIR sponsors.

References

1. Anguelov, D., Srinivasan, P., Koller, D., Thrun, S., Rodgers, J., Davis, J.: SCAPE: Shape Completion and Animation of PEople. ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH (2005)
2. Blanz, V., Vetter, T.: A morphable model for the synthesis of 3d faces. In: ACM SIGGRAPH (1999)
3. Cashman, T.J., Fitzgibbon, A.W.: What shape are dolphins? building 3D morphable models from 2D images. TPAMI (2013)
4. Choy, C.B., Xu, D., Gwak, J., Chen, K., Savarese, S.: 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In: ECCV (2016)
5. Cootes, T.F., Taylor, C.J.: Active shape modelssmart snakes. In: BMVC (1992)
6. Dürer, A.: Four Books on Human Proportion. Formschneyder (1528)
7. Fan, H., Su, H., Guibas, L.J.: A point set generation network for 3d object reconstruction from a single image. In: CVPR (2017)
8. Girdhar, R., Fouhey, D., Rodriguez, M., Gupta, A.: Learning a predictable and generative vector representation for objects. In: ECCV (2016)
9. Gwak, J., Choy, C.B., Garg, A., Chandraker, M., Savarese, S.: Weakly supervised 3d reconstruction with adversarial constraint. In: 3DV (2017)
10. Häne, C., Tulsiani, S., Malik, J.: Hierarchical surface prediction for 3d object reconstruction. In: 3DV (2017)
11. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: ICCV (2017)
12. He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks. In: ECCV (2016)
13. Hughes, J.F., Foley, J.D.: Computer graphics: principles and practice. Pearson Education (2014)
14. Kanazawa, A., Black, M.J., Jacobs, D.W., Malik, J.: End-to-end recovery of human shape and pose. In: CVPR (2018)
15. Kar, A., Tulsiani, S., Carreira, J., Malik, J.: Category-specific object reconstruction from a single image. In: CVPR (2015)
16. Kato, H., Ushiku, Y., Harada, T.: Neural 3d mesh renderer. In: CVPR (2018)
17. Khamis, S., Taylor, J., Shotton, J., Keskin, C., Izadi, S., Fitzgibbon, A.: Learning an efficient model of hand shape variation from depth images. In: CVPR (2015)
18. Laine, S., Karras, T., Aila, T., Herva, A., Saito, S., Yu, R., Li, H., Lehtinen, J.: Production-level facial performance capture using deep convolutional neural networks. In: Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation (2017)
19. Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., Black, M.J.: SMPL: A skinned multi-person linear model. ACM Trans. Graphics (Proc. SIGGRAPH Asia) (2015)
20. Pinkall, U., Polthier, K.: Computing discrete minimal surfaces and their conjugates. Experimental mathematics (1993)
21. Rezende, D.J., Eslami, S.A., Mohamed, S., Battaglia, P., Jaderberg, M., Heess, N.: Unsupervised learning of 3d structure from images. In: NIPS (2016)
22. Saito, S., Wei, L., Hu, L., Nagano, K., Li, H.: Photorealistic facial texture inference using deep neural networks. In: CVPR (2017)
23. Sela, M., Richardson, E., Kimmel, R.: Unrestricted facial geometry reconstruction using image-to-image translation. In: ICCV (2017)
24. Sinha, A., Unmesh, A., Huang, Q., Ramani, K.: Surfnet: Generating 3d shape surfaces using deep residual networks. In: CVPR (2017)
25. Sorkine, O., Cohen-Or, D., Lipman, Y., Alexa, M., Rössl, C., Seidel, H.P.: Laplacian surface editing. In: Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing. pp. 175–184. ACM (2004)

26. Tatarchenko, M., Dosovitskiy, A., Brox, T.: Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. In: ICCV (2017)
27. Taylor, J., Stebbing, R., Ramakrishna, V., Keskin, C., Shotton, J., Izadi, S., Hertzmann, A., Fitzgibbon, A.: User-specific hand modeling from monocular depth sequences. In: CVPR (2014)
28. Tewari, A., Zollhöfer, M., Kim, H., Garrido, P., Bernard, F., Pérez, P., Theobalt, C.: Mofa: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction. In: ICCV (2017)
29. Thompson, D.: On Growth and Form. Cambridge Univ. Press (1917)
30. Tulsiani, S., Zhou, T., Efros, A.A., Malik, J.: Multi-view supervision for single-view reconstruction via differentiable ray consistency. In: CVPR (2017)
31. Vicente, S., Carreira, J., Agapito, L., Batista, J.: Reconstructing pascal voc. In: CVPR (2014)
32. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The Caltech-UCSD Birds-200-2011 Dataset. Tech. Rep. CNS-TR-2011-001, California Institute of Technology (2011)
33. Wu, J., Wang, Y., Xue, T., Sun, X., Freeman, W.T., Tenenbaum, J.B.: MarrNet: 3D Shape Reconstruction via 2.5D Sketches. In: NIPS (2017)
34. Yan, X., Yang, J., Yumer, E., Guo, Y., Lee, H.: Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision. In: NIPS (2016)
35. Yang, B., Rosa, S., Markham, A., Trigoni, N., Wen, H.: 3d object dense reconstruction from a single depth view. arXiv preprint arXiv:1802.00411 (2018)
36. Yang, Y., Ramanan, D.: Articulated pose estimation with flexible mixtures-of-parts. In: CVPR (2011)
37. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep networks as a perceptual metric. In: CVPR (2018)
38. Zhou, T., Tulsiani, S., Sun, W., Malik, J., Efros, A.A.: View synthesis by appearance flow. In: ECCV (2016)
39. Zhu, R., Kiani, H., Wang, C., Lucey, S.: Rethinking reprojection: Closing the loop for pose-aware shape reconstruction from a single image. In: ICCV (2017)
40. Zuffi, S., Kanazawa, A., Jacobs, D., Black, M.J.: 3d menagerie: Modeling the 3d shape and pose of animals. In: CVPR (2017)