18/10/03

IVILAB meeting notes

Announcements

- Notebooks
  - I am happy to get them in reverse chronological order if that works better for you
  - Just include “reverse chronological order” on the title page
  - One way is to put it into the latex \data{} macro
- Paper study format
  - Presentation should be more or less self contained
    - Assume that many have not been able to look at it
    - We will have a few interested parties commit to reading it as well to increase the possibility of interesting discussion.
- Connecting reading and writing
  - E.g., notice that the figure captions in this paper are generally helpful.

18/10/03

Learning meshes for objects

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/.

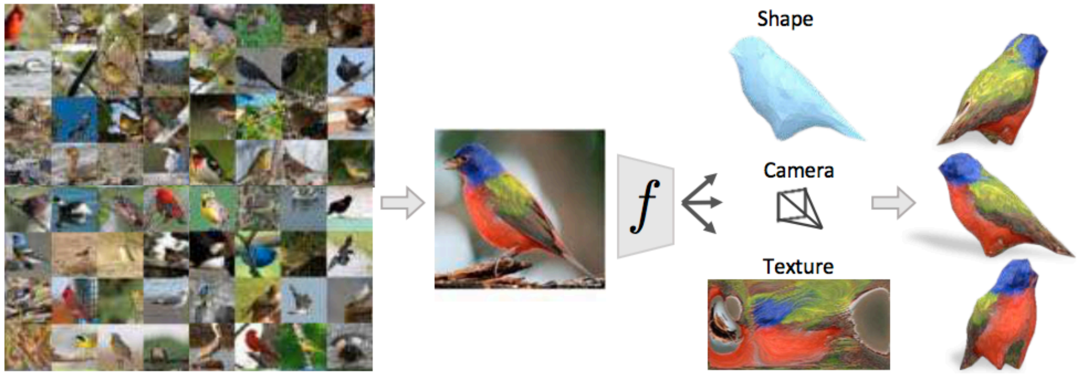

Fig. 1: Given an annotated image collection of an object category, we learn a predictor f that can map a novel image I to its 3D shape, camera pose, and texture.

Main ideas

- CNN to learn shape representations from images
  - What is a CNN?

- Shape representation is a fixed size mesh that is topologically a sphere
  - Entails silhouette via rendering
  - Includes semantic keypoints and where they get rendered
  - Includes a texture mapping (links to image)
  - Initialized in training to something reasonable

- Learn a weak perspective camera
  - f=\infinity

Mesh point indexing

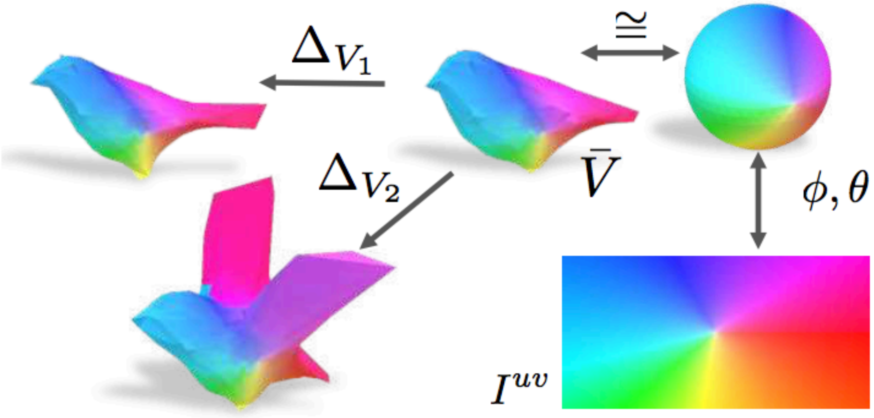

Fig. 3: **Illustration of the UV mapping.** We illustrate how a texture image I^{uv} can induce a corresponding texture on the predicted meshes. A point on a sphere can be mapped onto the image I^{uv} via using spherical coordinates. As our mean shape has the same mesh geometry (vertex connectivity) 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.

Main ideas (II)

- Mesh is category mesh + instance deformation

- 3D training data is hard to arrange, 2D less so
  - Use segmented birds with semantic keypoint labels
    - E.g, tip of beak

- Priors
  - Assume symmetry
  - Deformation should be small (regularized)
  - Mesh should be smooth
  - Semantic keypoint labels should be sparse (peaked distribution)

- Texture “flow” (mentioned in next figure, more later)

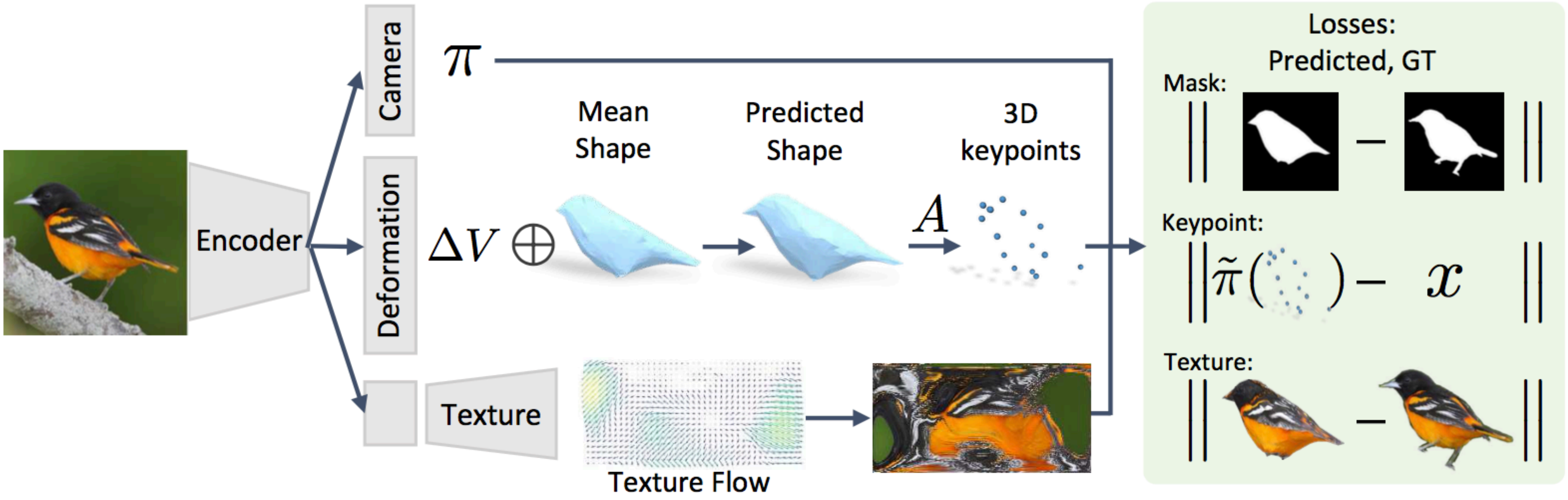

Fig. 2: **Overview of the proposed framework.** An image I is passed through a convolutional encoder to a latent representation that is shared by modules that estimate the camera pose, deformation and texture parameters. Deformation is an offset to the learned mean shape, which when added yield instance 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.

Texture “flow”

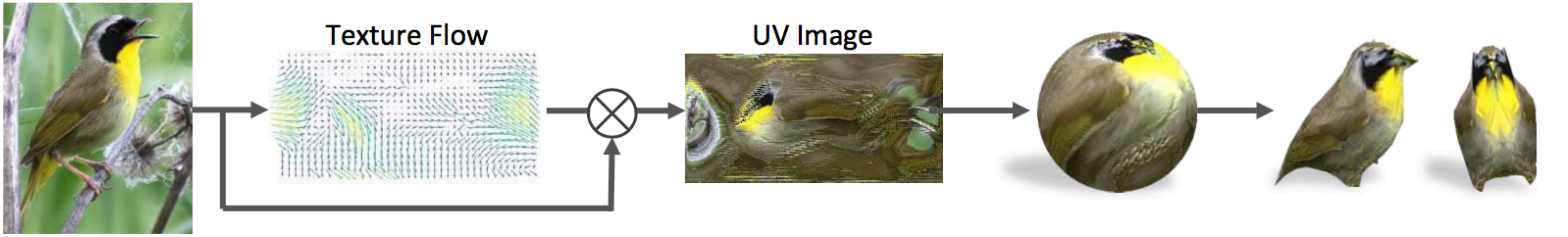

Fig. 4: **Illustration of texture flow.** We predict a texture flow $\mathcal{F}$ that is used to bilinearly sample the input image I to generate the texture image I^{uv}. We can use this predicted UV image I^{uv} to then texture the instance mesh via the UV mapping procedure illustrated in Figure 3.

Main idea for learning: Learn which pixel location in the image should be used for transfer.

Main ideas (learning)

- Learning needs everything to be differentiable
  - Projection, even weak projection, is not
    - Rasterization is a big issue

- They use a new fancy projection method (next slide)

Neural 3D Mesh Renderer

Hiroharu Kato[1], Yoshitaka Ushiku[1], and Tatsuya Harada[1,2]
[1]The University of Tokyo, [2]RIKEN
{kato,ushiku,harada}@mi.t.u-tokyo.ac.jp

Abstract

For modeling the 3D world behind 2D images, which 3D representation is most appropriate? A polygon mesh is a promising candidate for its compactness and geometric properties. However, it is not straightforward to model a polygon mesh from 2D images using neural networks because the conversion from a mesh to an image, or rendering, involves a discrete operation called rasterization, which prevents back-propagation. Therefore, in this work, we propose an approximate gradient for rasterization that enables the integration of rendering into neural networks. Using this renderer, we perform single-image 3D mesh reconstruction with silhouette image supervision and our system outperforms the existing voxel-based approach. Additionally, we perform gradient-based 3D mesh editing operations, such as 2D-to-3D style transfer and 3D DeepDream, with 2D supervision for the first time. These applications demonstrate the potential of the integration of a mesh renderer into neural networks and the effectiveness of our proposed renderer.

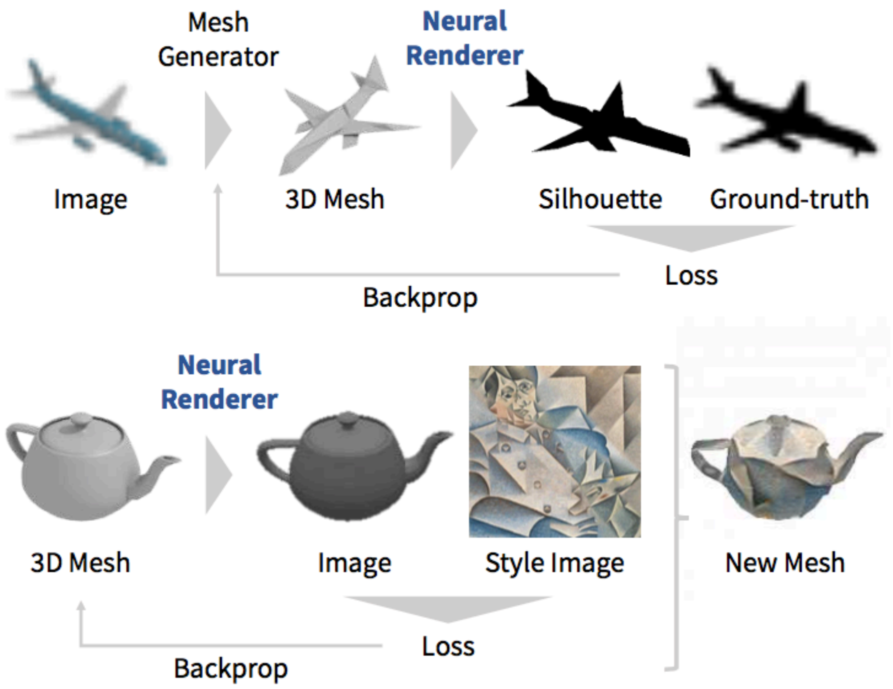

Figure 1. Pipelines for single-image 3D mesh reconstruction (upper) and 2D-to-3D style transfer (lower).

are 3D extensions of pixels, are the most widely used format in machine learning because they can be processed by

Results

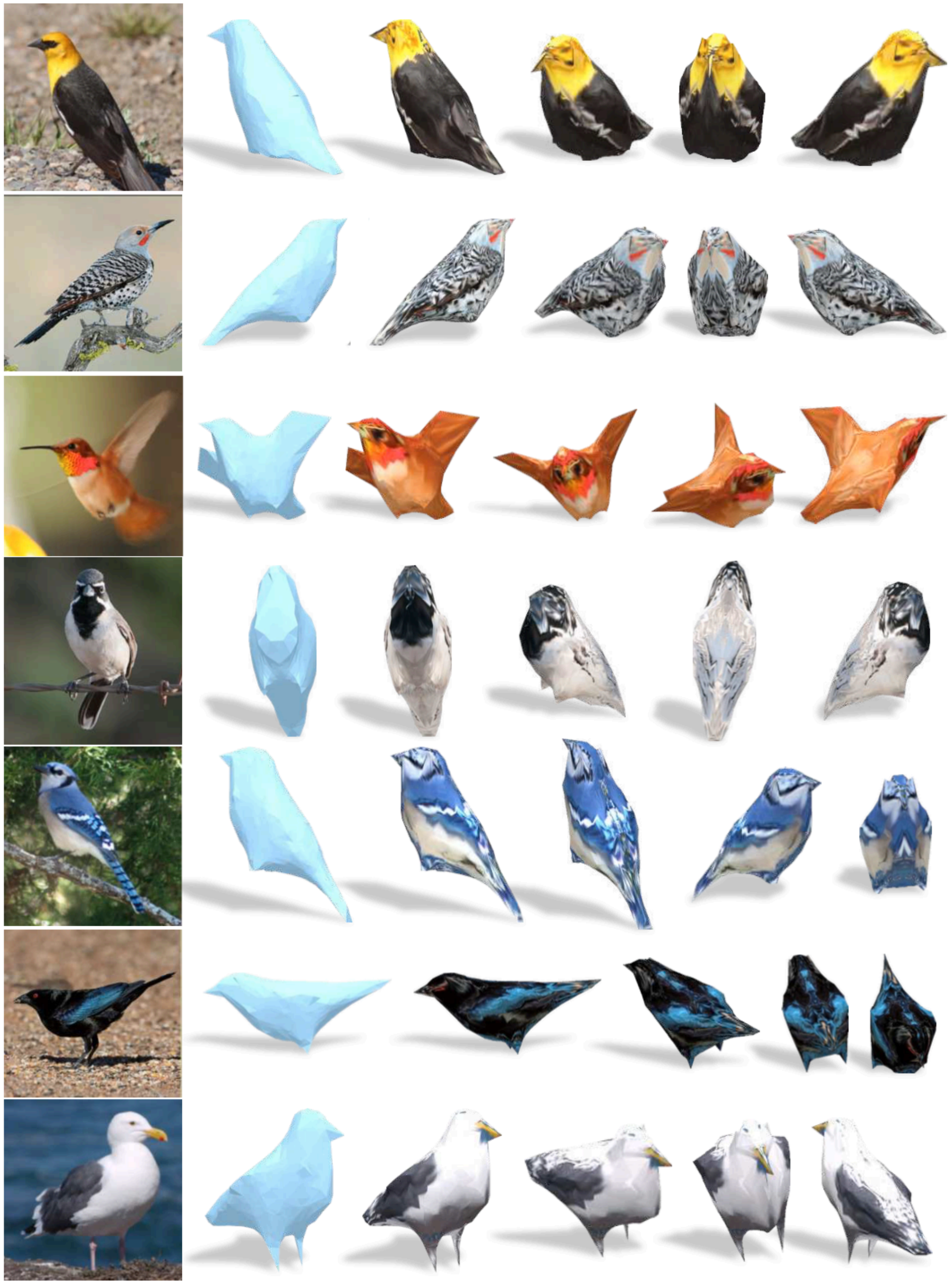

Fig. 5: **Sample results.** We show predictions of our approach on images from the test set. For each input image on the left, we visualize (in order): the predicted 3D shape and texture viewed from the predicted camera, and textured shape from three novel viewpoints. See the appendix

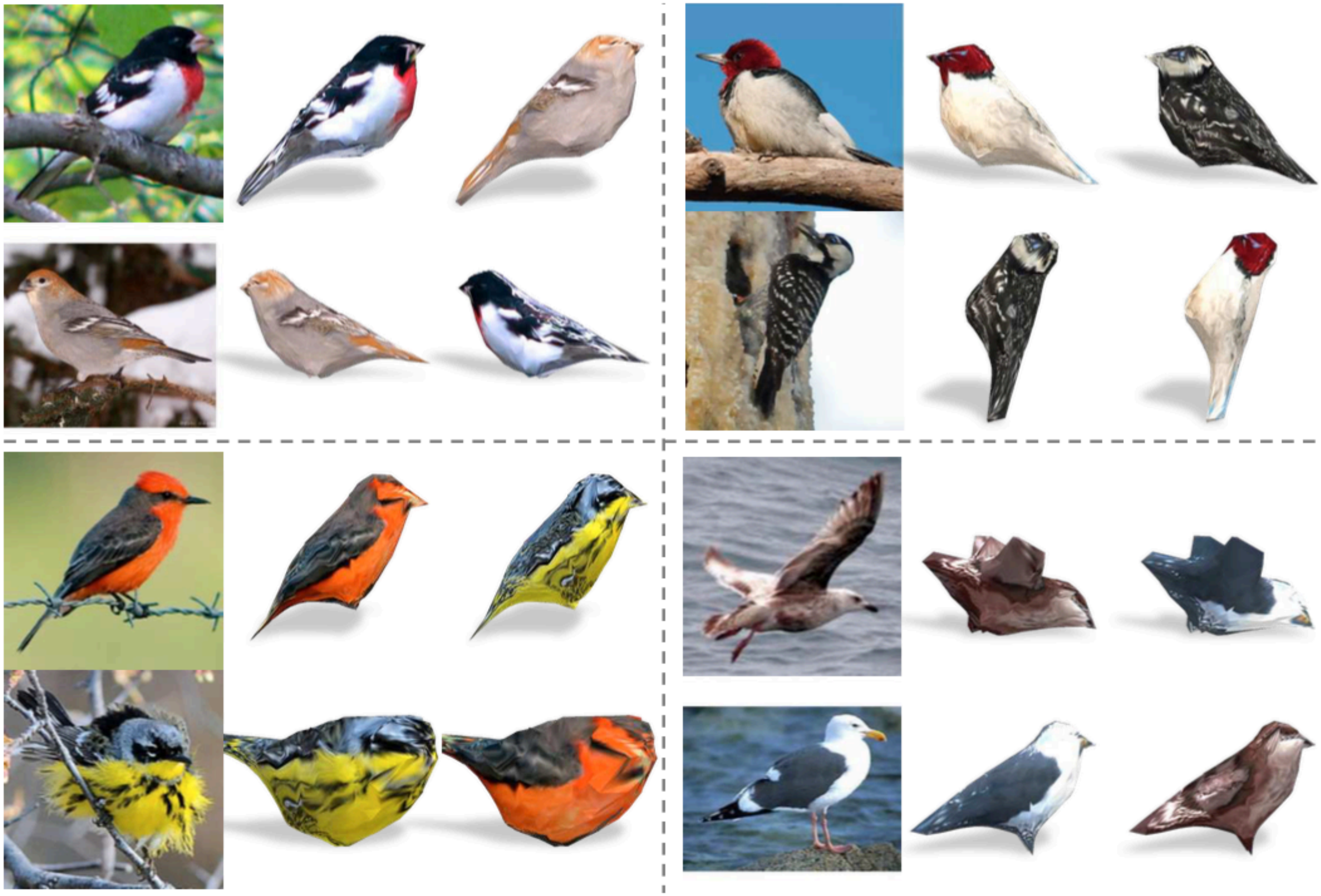

Fig. 7: **Texture Transfer Results.** Our representation allows us to easily transfer the predicted texture across instances using the canonical appearance image (see text for details). We visualize sample results of texture transfer across different pairs of birds. For each pair, we show (left): the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

Results

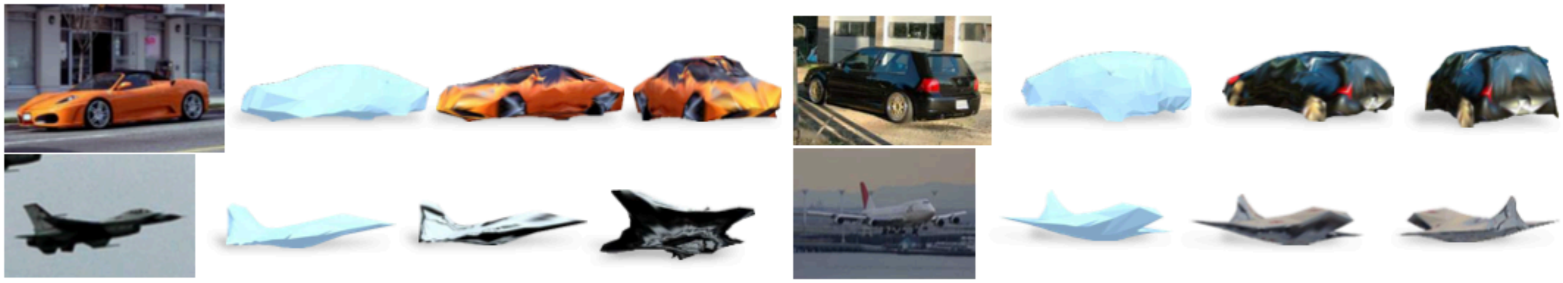

Fig. 9: **Pascal 3D+ results.** We show predictions of our approach on images from the test set. For each input image on the left, we visualize (in order): the predicted 3D shape viewed from the predicted camera, the predicted shape with texture viewed from the predicted camera, and the shape with texture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