Canonical Surface Mapping via Geometric Cycle Consistency

Nilesh Kulkarni  Abhinav Gupta*  Shubham Tulsiani*
Carnegie Mellon University  Facebook AI Research
{nileshk, abhinavg}@cs.cmu.edu  shubhtuls@fb.com
https://nileshkulkarni.github.io/csm/

Figure 1: We study the task of Canonical Surface Mapping (CSM). This task is a generalization of keypoint estimation and involves mapping pixels to canonical 3D models. We learn CSM prediction without requiring correspondence annotations, by instead using geometric cycle consistency as supervision. This allows us to train CSM prediction for diverse classes, including rigid and non-rigid objects.

Abstract

We explore the task of Canonical Surface Mapping (CSM). Specifically, given an image, we learn to map pixels on the object to their corresponding locations on an abstract 3D model of the category. But how do we learn such a mapping? A supervised approach would require extensive manual labeling which is not scalable beyond a few hand-picked categories. Our key insight is that the CSM task (pixel to 3D), when combined with 3D projection (3D to pixel), completes a cycle. Hence, we can exploit a geometric cycle consistency loss, thereby allowing us to forgo the dense manual supervision. Our approach allows us to train a CSM model for a diverse set of classes, without sparse or dense keypoint annotation, by leveraging only foreground mask labels for training. We show that our predictions also allow us to infer dense correspondence between two images, and compare the performance of our approach against several methods that predict correspondence by leveraging varying amount of supervision.

1. Introduction

Plato famously remarked that while there are many cups in the world, there is only one ‘idea’ of a cup. Any particular instance of a category can thus be understood via its relationship to this platonic ideal. As an illustration, consider an image of a bird in Figure 1. When we humans see this image, we can not only identify and segment the bird but also go further and even map pixels to an abstract 3D representation of the category. This task of mapping pixels in an image to locations on an abstract 3D model (which we henceforth call canonical surface mapping) is generalization and densification of keypoint estimation and is key towards rich understanding of objects. But how do we learn to do this task? What is the right data, supervision or models to achieve dense rich understanding of objects?

One way to learn the canonical surface mapping task is to collect large-scale labeled data. Specifically, we can label hundreds or thousands of keypoints per image for thousands

* the last two authors were equally uninvolved.
of images. As each keypoint location defines which pixel corresponds to a specific location on the 3D surface, this approach of manually labeling the keypoints can provide dense supervision for learning. This approach has in fact been shown to be quite successful for specific categories such as humans [2]. But of course collecting such labeled data requires enormous manual labeling effort, making it difficult to scale to generic categories.

Is there an alternative supervisory signal that can allow one to learn without reliance on such labeled data? Interestingly, we note that this task of canonical surface mapping is an inverse graphics task. Any such mapping is constrained by the geometry operating on the underlying 3D, and any predicted mapping should also respect this structure. In particular, for the pixels that belong to the object given by the object mask, the CSM function maps these pixels onto the 3D shape. These points on the 3D shape, when projected back using (known/predicted) camera, should map back to the same pixels. Our key insight is that one can complete the cycle (pixels $\rightarrow$ 3D $\rightarrow$ pixels) and use the consistency loss as an objective. The gradients from the loss can be propagated back to the CSM function prediction function, thereby allowing us to learn this mapping without reliance on strong forms of supervision.

In this paper, we present an approach to learn the task of canonical surface mapping from the set of images belonging to semantic category, their input masks and an abstract 3D model which represents the semantic category. Additionally, we show that predicting a canonical surface mapping for images allows us to infer dense correspondence across images of a category, and our approach enables recovery of dense correspondences without any correspondence supervision! In comparison to approaches that use dense supervision for this task [2], or approaches that leverage keypoints for the related tasks of semantic correspondence [7], or 3D reconstruction [18], this is significant decrease in supervision. This allows us to train our CSM model for a diverse set of classes: birds, zebras, cars and more (See Figure 1). We believe our approach can pave the way for large-scale internet-driven 3D understanding and correspondence inference since both semantic imagesets and masks are easy to obtain (and automatic approaches can be used as well).

2. Related Work

Dense Semantic Correspondences. A fundamental task that is equivalent to pursuing canonical surface mapping is that of inferring dense semantic correspondence – given two images, the goal is to predict for each pixel in the former, the corresponding pixel in the latter. Methods prior to the recent resurgence of deep learning [22, 24] demonstrated that matching using features such as SIFT could allow recovering correspondence across instances, and later work showed similar results using CNN features [13, 25]. While these generic features allow recovering correspondence, learning specifically for the task using annotated data can improve results [7]. However, collecting such annotation can be tedious, so several approaches have attempted to relax the supervision for learning correspondence.

Among these, a common paradigm is to learn correspondence by self-supervision, where random perturbations of images are used as training pairs. This allows predicting parametric warping [17, 31, 32] to relate images, or learn equivariant embeddings [38] for matching. However, the these methods are fundamentally restricted to training data of the same instance, with no change in the visible content, thereby limiting the performance for different instances with viewpoint changes. While for certain categories of interest *e.g.* humans, some approaches [27, 30, 35, 36, 42] show that it is possible to use calibrated multi-view or motion capture to generate supervision, this form of supervision is slightly tedious to collect for all classes. An alternate form of supervision can come via synthetic data, where synthetic image pairs rendered using the same pose as a real image pair, can help learn a correspondence function between real images that is cycle-consistent [52]. However, this approach relies on availability of large-scale synthetic data and known pose for real images to generate the supervisory signal, and we show that both these requirements can be relaxed.

Learning Invariant Representations. Our work is broadly related to methods that learn pixel embeddings invariant to certain transforms. These approaches leverage tracking to obtain correspondence labels, and learn representations invariant to viewpoint transformation [34, 49] or motion [44]. Similar to self-supervised correspondence approaches, these are also limited to training using observations of the same instance, and do not generalize well across instances. While our canonical surface mapping is also a pixel-wise embedding invariant to certain transforms, it has a specific geometric meaning *i.e.* correspondence to a 3D surface, and leveraging this is what allows learning without the correspondence supervision.

Category-Specific 3D Reconstruction. A related line of work pursued in the community is that of reconstructing the instances in a category using category-specific deformable models. Dating back to the seminal work of Blanz & Vetter [4], who operationalized D’Arcy Thompson’s insights into the manifold of forms [39], morphable 3D models have been used to model faces [4], hands [21, 37], humans [3, 26] and other generic classes [5, 18, 19, 46].

In conjunction with known/predicted camera parameters, this representation also allows one to extract a pixelwise canonical mapping. However, these methods often rely on 3D training data to infer this representation. Even approaches that relax this supervision [18, 19, 46] crucially
rely on (sparse or dense) 2D keypoint annotations during training. In contrast, we show that learning a canonical surface mapping is feasible even without such supervision. Further, we demonstrate that directly learning the mapping function leads to more accurate results than obtaining these via an intermediate 3D estimate.

**Consistency as Meta-Supervision.** Ours is not the only task where acquiring direct supervision is often infeasible, and the idea of leveraging some form of consistency to overcome this hurdle has been explored in several domains. Recent volumetric reconstruction \[12, 29, 41, 48\] or depth prediction \[10, 11, 50\] approaches use geometric consistency between the predicted 3D and available views as supervision. Similarly, the notion that when learning some transformations, their composition often respects a cyclic structure has been used for image generation \[23, 53\], correspondence estimation \[50, 51\] etc. In our setup, we also observe that the approach of using consistency as meta-supervision allows bypassing supervision. We do so by leveraging insights related to both, geometry and cycle consistency – given a surface mapping, there is a geometrically defined inverse transform with which the canonical surface mapping predictions should be cycle-consistent.

### 3. Approach

Given an image, our goal is to infer for each pixel on the object, its mapping onto a given canonical template shape of the category. We do so by learning a parametrized CNN \(f_\theta\), which predicts a pixelwise canonical surface mapping (CSM) given an input image. We show that our method, while only relying on foreground masks as supervision, can learn to map pixels to the given category-level template shape. Our key insight is that this mapping function we aim to learn has a geometric structure that should be respected by the predictions. We operationalize this insight, and learn a CSM predictor using a geometric cycle consistency loss, thereby allowing us to bypass the need for supervision in the form of annotated (sparse or dense) keypoints.

We first present in Section 3.2 our training setup in a scenario where the camera pose for each training image is given. We then show how we can relax this requirement of known camera in Section 3.3. Learning a CSM predictor implicitly allows us to capture the correspondence across instances, and we describe in Section 3.4 the procedure to recover dense semantic correspondence given two images.

#### 3.1. Preliminaries

**Surface Parametrization.** The template shapes we learn mappings to are in fact two-dimensional surfaces in 3D space. The surface \(S\) of the template shape can therefore be parametrized via two parameters \(u \in (0, 1)\) and \(v \in (0, 1)\) (or equivalently a 2D vector \(\mathbf{u}\)). This parametrization implies that we can obtain a mapping \(\phi\) such that \(\phi(\mathbf{u})\) represents a unique point on the surface \(S\).

![Figure 2: Surface Parametrization. We show the mapping from \((u, v)\) space to the surface of the 3D model for two categories.](image)

While there are several ways to construct such a mapping, one intuitive way is to consider \(\mathbf{u}\) to represent the polar angles to parametrize points on the surface of a hollow sphere, which can be mapped to a surface \(S\) by pushing it inward \[28\]. Given a template shape with a surface \(S\), we use this approach to obtain the parametrization \(\phi\). We show some visualizations in Figure 2 for the mapping from a 2D square to template 3D shapes for two categories.

**Canonical Surface Mapping.** A canonical surface mapping \(C\) for an image \(I\) is a mapping from pixels onto the template 3D shape. Given a pixel \(p \equiv (x, y)\), \(C[p]\) represents the corresponding point on the surface. As the surface has a two-dimensional parametrization, \(C\) is equivalently an image of the same size as \(I\), with a two-channel value at each pixel. Our parametrized CNN \(f_\theta\) that predicts this mapping from an input image, therefore learns a per-pixel prediction task – given an RGB input image, it outputs a 2 dimensional vector for each pixel.

**Camera Projection.** We model the camera as a weak perspective (scaled orthographic) transformation. We represent the camera for every image \(I\) as \(\pi\), parameterized by the scale \(s \in \mathcal{R}\), translation \(t \in \mathcal{R}^2\) and rotation \(r\) are three euler angles. We denote by \(\pi(P)\) as the projection of a point \(P\) to the image coordinate frame using the camera parameters \(\pi \equiv (s, t, r)\).

#### 3.2. Learning via Geometric Cycle Consistency

We aim to learn a per-pixel predictor \(f_\theta\) that outputs a canonical surface mapping given an input image \(I\). We present an approach to do so using only foreground masks as supervision. However, for simplicity, we first describe here how we can learn this CSM predictor assuming known camera parameters for each training image, and relax this requirement in Section 3.3. Our approach is to derive learning signal from the geometric
Incorporating Visibility Constraints. Enforcing that the pixels when lifted to 3D, project back to the same location is desirable, but not a sufficient condition. As an illustration, for a front facing bird, both the beak and tail project at similar locations, but only the former would be visible. This implies that points on the surface that are self-occluded under π can also result in minimizing $L_{cyc}$. Our solution is to discourage $f_\theta$ from minimizing $L_{cyc}$. Our solution is to discourage $f_\theta$ from self-occluded regions under camera π. A point on the 3D shape is self-occluded under a camera π, its z-coordinate in camera frame is larger than the rendered depth at the corresponding pixel. We use Neural Mesh Renderer (NMR) [20] to render a depth map $D_\pi$ for the template shape $S$ under camera π, and define a visibility loss for each pixel $p$ by checking if the z-coordinate (say $z_p$) of its corresponding point $\phi(C[p])$ on the 3D shape, when projected under π, has a larger z-coordinate.

$$L_{vis} = \sum_{p \in I_f} \max(0, z_p - D_\pi(\bar{p}))$$

Network Details. We implement $f_\theta$ as a network with UNet [33] style architecture. This network takes as input an image of size 256 x 256 and outputs a unit vector per pixel representing a point on surface of sphere which is then converted to a $(u, v)$ coordinate analogous to latitude and longitude. We train our network to minimize the cycle-consistency and visibility objectives:

$$L_{consistency} = L_{vis} + L_{cyc}$$

Even though we do not have direct supervision for the mappings, as we train a shared predictor across instances, the explicit priors for geometric consistency, and the implicit inductive biases in CNNs for spatial equivariance are sufficient for us to learn a meaningful predictor.

Foreground Mask Prediction. While the training procedure described above encourages cycle-consistent predictions at pixels belonging to the object, the learned CNN $f_\theta$ also predicts some (possibly spurious) values at other pixels. To allow us to ignore these background pixels for inferring correspondence (see Section. 3.4), as well as for generating visualizations, we train an additional per-pixel mask predictor using standard cross-entropy loss $L_{fg}$ against the ground-truth masks. To do so, we simply modify $f_\theta$ to yield an additional per-pixel foreground probability as output.
3.3. Learning without Pose Supervision

We have presented our approach to learn a canonical surface mapping predictor $f_\theta$ assuming known cameras $\pi$ for each training image. We note that our training objective is also differentiable w.r.t. the camera parameters, and we can therefore simply use predicted cameras instead of known cameras, and jointly learn pose and CSM prediction. This joint training can allow us to bypass the requirement of even camera supervision, and learn CSM prediction using only foreground mask annotations and a given template shape. We therefore learn an additional camera-prediction CNN $g_\theta$, and use the predicted cameras to learn the CSM predictor via the geometric consistency training objectives. However, to overcome certain trivial solutions, we also add a mask reprojection error, and following [16, 40] use a multi-hypothesis camera predictor to avoid local minima. Our overall training setup is depicted in Figure 4.

Mask Re-projection Loss. If the only learning objective comprises of the self-consistency between camera predictions and the predicted CSMs, the networks can learn some trivial solutions e.g. always predict a ‘frontal’ camera and corresponding CSM. To avoid this we enforce that the the template shape, when viewed under a predicted camera $\pi$, should approximately match the known foreground image $I_{fg}$. To implement this loss, we use (NMR) [20] to obtain a differentiable render $f_{\text{render}}$, that given the template shape $S$ and a camera $\pi$, renders a mask. While the poses may still be ambiguous e.g. front and back facing cars, this additional mask reprojection loss allows us to circumvent the mentioned trivial solutions. This reprojection loss is defined as follows:

$$L_{\text{mask}} = \| f_{\text{render}}(S, \pi) - I_f \|^2$$  \hspace{1cm} (4)

Multi-Hypothesis Pose Prediction. Instead of predicting a single camera $\pi \equiv g_\theta(I)$, we follow previous methods [16, 40] and predict multiple hypotheses to overcome local minima. Our pose predictor outputs $\{(\pi_i, c_i) \} \equiv g_\theta(I)$ - a set of $N_c = 8$ pose hypotheses $\pi_i$, each with an associated probability $c_i$. We initialize the camera predictor $g_\theta$ using a pre-trained ResNet-18 network [15].

Overall Training Objective. As our pose predictor yields multiple pose hypotheses $\pi_i$, each with an associated probability $c_i$, we can train our networks by minimizing the expected loss. We denote by $L_{\text{cyc}}^i, L_{\text{vis}}^i, L_{\text{mask}}^i$ the corresponding losses under the camera prediction $\pi_i$. In addition to minimizing the expected loss over these terms, we also use an additional diversity prior $L_{\text{div}}$ to encourage diverse pose hypotheses (see appendix for details). The overall training objective using these is:

$$L_{\text{tot}} = L_{\text{div}}(g_\theta(I)) + \sum_{i=1}^{N_c} c_i (L_{\text{cyc}}^i + L_{\text{vis}}^i + L_{\text{mask}}^i)$$ \hspace{1cm} (5)

This framework allows us to learn the canonical surface mapping function $f_\theta$ via geometric cycle consistency, using only foreground mask annotations in addition to the given template shape. Once the network $f_\theta$ is learned, we can infer a canonical surface map from any unannotated image.

3.4. Dense Correspondences via CSM

We described an approach for predicting canonical surface mappings without relying on pose or keypoint annotations. This allows us to infer dense semantic correspondences given two images of the same semantic object category, because if pixels across images correspond, they should get mapped to the same region on the canonical surface. Given a (source, target) image pair $(I_s, I_t)$, let us denote by $(C_s, I_{fg}, I_{fg})$ the corresponding predicted canonical surface mappings and foreground masks. Given these predictions, for any pixel $p_i$ on $I_s$, we can infer its corresponding pixel $T_{s\rightarrow t}[p_i]$ on $I_t$ by searching for the (foreground) pixel that maps closest to $\phi(C_s[p_i])$:

$$T_{s\rightarrow t}[p_i] = \arg \min \limits_{p_j \in I_{fg}} \| \phi(C_s[p_i]) - \phi(C_t[p_j]) \|$$ \hspace{1cm} (6)

Not only does our approach allow us to predict correspondences for pixels between two images, it also allows us to infer regions of non-correspondence i.e. pixels in source image for which correspondences in the target image do not exist (e.g. most pixels between a left and right facing bird do not correspond). We can infer these by simply denoting pixels for which the minimum distance in Eq. 6 is above a certain threshold as not having a correspondence in the target image. This ability to infer non-correspondence is particularly challenging for self-supervised methods that generate data via random warping [17, 31, 38] as the training pairs for these never have non-corresponding regions.

4. Experiments

Our approach allows us to predict canonical surface mappings across generic categories. However, due to lack of annotation for the task, which is in fact our motivation for learning without supervision, it is difficult to directly evaluate the predictions. Instead, as our approach also allows us to recover correspondences across any two images (Section 3.4), we can evaluate these using the task of keypoint transfer. This is a well-studied task by approaches that learn semantic correspondence, and we report comparisons to baselines that leverage varying degree of supervision while training. We first report these comparisons in Section 4.1, and then present results for additional generic categories (e.g. horses, sheep, cows) in Section 4.2, using Imagenet images with automatically obtained segmentation masks.

4.1. Evaluation via Keypoint Transfer

We use our learned CSM prediction models for the task of keypoint transfer – given a source and target image pair,
where the source image has some annotated keypoints, the goal is to predict the location of these keypoints in the target image. We first describe the datasets used to train our model, and then briefly survey the various baselines we compare to and then present the evaluation results.

4.1.1 Experimental Setup

Datasets. We use bird images from the CUB-200-2011[43] and the car images from PASCAL3D+ [47] dataset for quantitative evaluation. CUB-200-2011 contains 6000 training and test images with 200 different species. Each bird has 14 annotated keypoints, a segmentation mask, and a bounding box. Note that we only use keypoint annotation at test time to evaluate our method on the task of dense correspondence as described earlier. We also train a model on the car category from PASCAL3D+ [47] which has over 6000 training and test images but evaluate only on cars from PASCAL VOC [9] with 12 keypoint annotations per instance. We downloaded a freely available mesh from [1] to serve as a bird template shape, used an average of 10 Shapenet [6] models to obtain a template shape for cars.

Baselines. We report comparisons to several methods that leverage varying amount of supervision for learning:

Category Specific Mesh Reconstruction (CMR) [18] learns to reconstruct the 3D shape and predict pose for a given instance, but relies on training time supervision of known keypoint locations and segmentation masks. Since a common morphable model is used across a category, we can compute the implied surface mappings via computing for each pixel, the coordinate of the mean shape that is rendered at its location (or nearest location in case of imperfect projection). We can then infer correspondences as in Section 3.4.

Zhou et al. [52] exploit a large collection of 3D synthetic models to learn dense correspondence via cycle-consistency. During training, they crucially rely on pose supervision (from PASCAL 3D+), as each cycle consists of synthetic images rendered from the same view as the real image pair. Their method outputs dense correspondences in the form of a per-pixel flow, and infers non-correspondence using a ‘matchability’ score.

Dense Equivariance (DE) [38] is a self-supervised method to learn correspondences, and does not require any pose or keypoint annotations. We re-implement this baseline such that it can exploit the annotations for object masks (see appendix for details). DE learns a per-pixel feature vector, and enforces corresponding pixels to have a similar feature. The supervision for correspondences is obtained via applying known in-plane random warps to images. During inference, we can recover the correspondence for a source pixel by searching for the most similar feature in the target image.

VGG Transfer. Inspired by Long et al.’s [25] observation that generic learned features allow recovering correspondences, we designed a baseline which infers correspondence via nearest neighbours in this feature space. Specifically for a pixel in the source image we lookup its VGG feature from the conv4 layer and finds its corresponding nearest neighbour in the target image (we found these features to perform better than AlexNet used by Long et al. [25]).

4.1.2 Evaluation Metrics

We evaluate the various methods on two metrics: a) Percentage of Correct Keypoints (PCK), and b) Keypoint Transfer AP (APK). We use two separate metrics, because while the PCK metric evaluates the accuracy of keypoint transfer for keypoints that are visible in both, source and target image, it does not disambiguate if an approach can infer that a particular source keypoint does not correspond to any pixel on the target. So therefore, while PCK lets us evaluate correspondence accuracy, the APK metric also lets us measure accuracy at inferring non-correspondence.

Percentage of Correct Keypoints (PCK): Given a (source, target) image pair with keypoint annotations on the source,
Figure 6: Predicted Canonical Surface mapping for six different categories. The color at each image pixel depicts the color at the corresponding surface point on the 3D template shape in the left row. While the predictions are mostly accurate, some error modes include: a) inferring globally incorrect CSM due to pose ambiguity (e.g. third horse), or b) incorrect local predictions due to missing segmentation (e.g. the second sheep).

Each method predicts a single estimate for the corresponding location in the target image. The PCK metric reports the mean accuracy of keypoint predictions across keypoints that are common across pairs. A prediction is considered correct only when the predicted location lies within $\alpha \times \max(h, w)$ radius around the ground truth annotation for the transfer. We report results with $\alpha = 0.1$, and $h, w$ refer to height and width of the image to which the keypoints were transferred.

Keypoint Transfer AP (APK): In addition to predicting a location in target image for each each keypoint in source image, this metric requires a confidence in the estimate. Ideally, if a source keypoint does not correspond in a target image, the corresponding predicted confidence should be low, whereas it should be high in case of a keypoint visible in both. Our approach and CMR [18] can rely on the (inverse) distance on the template/mean shape as a confidence measure. Zhou et al. [52] produce a ‘matchability’ score, and the feature based methods ‘DE’ [38] and ‘VGG transfer’ [25] can leverage feature similarity as confidence.

Given these predictions, we vary the confidence thresholds,
and observe that we learn accurate mappings and pose supervision (denoted as ‘CSM w/ pose’) to better compare to baselines that use similar approaches when using pose supervision. The area under the curve is reported in the legend for each of the CMU V isual and Robot Learning group and the one on the right is on cars and keypoints from PascalVOC [9]. See Section 4.1.2 for metric descriptions.

Table 1: PCK and APK. Percentage of correct keypoints (PCK) and Keypoint Transfer AP (APK) at \( \alpha = 0.1 \). See Section 4.1.2 for metric descriptions. All evaluations are on 10000 image pairs per category. Higher is better.

<table>
<thead>
<tr>
<th>Annotation</th>
<th>Method</th>
<th>Birds PCK</th>
<th>Cars PCK</th>
<th>Birds APK</th>
<th>Cars APK</th>
</tr>
</thead>
<tbody>
<tr>
<td>KP + Seg. Mask</td>
<td>CMR [18]</td>
<td>47.3</td>
<td>22.4</td>
<td>44.1</td>
<td>16.9</td>
</tr>
<tr>
<td>Pose + Syn. Data</td>
<td>Zhou et. al. [52]</td>
<td>-</td>
<td>-</td>
<td>37.1</td>
<td>10.5</td>
</tr>
<tr>
<td>Pose + Seg. Mask</td>
<td>CSM (ours) w/ pose</td>
<td>56.0</td>
<td>30.6</td>
<td>51.2</td>
<td>21.0</td>
</tr>
<tr>
<td>Seg. Mask</td>
<td>Dense Equi [38]</td>
<td>34.8</td>
<td>11.1</td>
<td>31.5</td>
<td>5.7</td>
</tr>
<tr>
<td></td>
<td>VGG Transfer</td>
<td>17.2</td>
<td>2.6</td>
<td>11.3</td>
<td>0.6</td>
</tr>
<tr>
<td></td>
<td>CSM (ours)</td>
<td>48.0</td>
<td>22.4</td>
<td>40.0</td>
<td>11.0</td>
</tr>
</tbody>
</table>

Figure 7: Keypoint Transfer PR Curves. We report the transfer precision vs recall curves for all the methods on the task of keypoint transfer. Dashed lines represent methods with pose or keypoint supervision. Solid lines denote approaches without such supervision. The area under the curve is reported in the legend for each of the plots (higher is better). The plot on left is for CUBS-Birds [43], and the one on the right is on cars and keypoints from PascalVOC [9]. See Section 4.1.2 for metric descriptions.

4.2. Learning from Unannotated Image Collections

As our method does not require keypoint supervision during training, we can apply it to learn canonical surface mappings for generic classes using just category-level image collections (with automatically obtained segmentation). We use images for various categories from ImageNet [8], obtain instance segmentation using an off-the-shelf system [14], and manually filter out instances with heavy occlusion. This results in about 1000 instances per category, and we train our CSM predictors using a per-category template model downloaded from the web (in fact, for zebras we use a horse model). We show qualitative results (on held-out images) in Figure 6 and observe that we learn accurate mappings that also respect correspondence across instance. Please see supplementary for additional visualizations.

5. Discussion

We present an approach to learn canonical surface mappings for generic categories using a geometric cycle consistency objective. Our approach allows us to do so without keypoint or pose supervision, and learn CSM prediction and infer dense correspondence while only relying on foreground masks as supervision. While this is an encouraging step towards understanding the underlying 3D structure and associations across images, several challenges still remain. In particular, as we seek to explain the per-pixel predictions via a reprojection of a single rigid template, our approach is not directly applicable to categories where the shapes across instances differ significantly or undergo large articulation. It would be interesting to extend our method to also allow for predicting the underlying deformation and articulation in addition to camera transforms. Additionally, while our approach allowed relaxing correspondence supervision, it would be desirable to take a step further, and learn from unannotated image collections without foreground mask supervision. Lastly, our approach leveraged geometric cycle consistency, and videos may provide an additional learning signal by enforcing consistency of predictions through time [45].

Acknowledgements. We would like the thank the members of the CMU Visual and Robot Learning group and the anonymous reviewers for helpful discussions and feedback. ST would also like to thank Alyosha Efros for his constant nudges to ‘work on cycles!’.

2209
References


