Cross-Domain 3D Equivariant Image Embeddings

Carlos Esteves [*] [1] Avneesh Sud [2] Zhengyi Luo [1] Kostas Daniilidis [1] Ameesh Makadia [2]

Abstract

Spherical convolutional networks have been introduced recently as tools to learn powerful feature representations of 3D shapes. Spherical CNNs are equivariant to 3D rotations making them ideally suited to applications where 3D data may be observed in arbitrary orientations. In this paper we learn 2D image embeddings with a similar equivariant structure: embedding the image of a 3D object should commute with rotations of the object. We introduce a cross-domain embedding from 2D images into a spherical CNN latent space. This embedding encodes images with 3D shape properties and is equivariant to 3D rotations of the observed object. The model is supervised only by target embeddings obtained from a spherical CNN pretrained for 3D shape classification. We show that learning a rich embedding for images with appropriate geometric structure is sufficient for tackling varied applications, such as relative pose estimation and novel view synthesis, without requiring additional task-specific supervision.

1. Introduction

The success of CNNs in computer vision has shown that large training datasets and task-specific supervision are sufficient to learn rich feature representations for a variety of tasks such as image classification and object detection (He et al., 2016). However, there remain many challenges, such as motion estimation and view synthesis, which require complex geometric reasoning and for which labeled data is not available at scale. For such problems there is a trend towards developing models with geometry-aware latent representations that can learn the structure of the world without requiring full geometric supervision (e.g. Kulkarni et al. (2015); Rhodin et al. (2018); Yang et al. (2015); Yan et al. (2016); Mahjourian et al. (2018); Eslami et al. (2018)).

A desirable property for an image embedding is robustness to 3D geometric transformations of the scene. 3D rotations, in particular, are a nuisance to computer vision algorithms because even small 3D rotations of objects in the world can induce large transformations in image space. In recent years there has been much attention given to the study of equivariant neural networks (e.g. Cohen & Welling (2016); Worrall et al. (2017a)), as equivariant maps provide a natural formulation to address group transformations on images. Despite these advances, designing a 3D rotation equivariant map of 2D images is an open challenge. This is because the rotation of a 3D object does not act directly on the pixels of the resulting image due to the intervening camera projection. Thus, an equivariant map cannot be constructed by design and instead an (approximate) equivariant map must be learned. This is the central task of the paper: *how can we learn an embedding for images of 3D objects that is equivariant to 3D rotations of the objects?*

Our proposal is inspired by recent works on 3D rotation equivariant CNNs for 3D shape representations (Cohen et al., 2018; Esteves et al., 2018). These works show that spherical convolutional networks can achieve state of the art performance on 3D shape classification and pose estimation tasks, and these networks' equivariance properties mean their performance does not suffer when considering 3D shapes in arbitrary orientations.

In this work we propose to learn an equivariant embedding of an image by mapping it into the equivariant feature space of a spherical CNN trained on datasets of 3D shapes. Our approach is unique in that we are directly supervising the desired target embeddings with the pretrained 3D shape features and we do not consider any other task-specific training losses. By bootstrapping with features of 3D shapes, our model (1) encodes images with the shape properties of the observed object and (2) has an an underlying spherical structure that is equivariant to 3D rotations of the object.

The cross-domain embeddings can be used for different applications, either directly or indirectly, without requiring any additional task-specific supervised training. We illustrate this point by showing results on two very different challenges:

[*] Work done during an internship at Google. [1] GRASP Laboratory, University of Pennsylvania [2] Google Research. Correspondence to: Carlos Esteves <machc@seas.upenn.edu>.

Proceedings of the 36^th International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019. Copyright 2019 by the author(s).

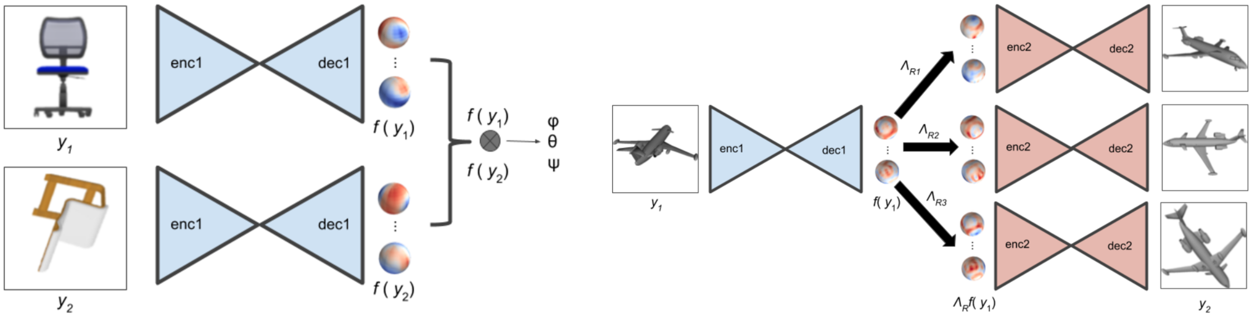

Figure 1. **Overview.** We learn category based spherical 3D equivariant embeddings that can be correlated for relative pose estimation, and rotated for novel view synthesis. *Left: relative pose estimation.* Given 2 images of objects from same class, we obtain the respective spherical embeddings. The relative pose is computed from the spherical correlation between the spherical embeddings. *Right: novel view synthesis.* We first embed the input view into the spherical representation, then we apply the target rotation to the spherical feature maps, and feed them to the synthesizer to generate novel views.

Relative orientation estimation Our model maps images to rotation equivariant embeddings defined on the sphere (Fig. 1-*left*). We can recover the relative orientation between two embeddings by finding the rotation that brings them into alignment. We do this simply with spherical cross-correlation without running any regression as an unsupervised spatial transformer (Jaderberg et al., 2015) would do. This method approaches state of the art even though it uses no task-specific training. The same method can be applied to align 2D images with 3D shapes.

Novel view synthesis The learned embeddings also encode enough shape properties to synthesize new views. By simply training a decoder from the spherical embedding space with a photometric loss, we have a model for novel view synthesis. New views are generated by rotating the latent embedding (Fig. 1-*right*). No task specific supervision (e.g. an image and its rotated counterpart) is required.

To reiterate, our main contribution is a novel cross-domain neural model that can map 2D images into a 3D rotation equivariant feature space. Generating spherical feature maps from 2D images is a complex high-dimensional regression task, mapping between topologies, which requires a novel encoder-decoder architecture. We consider the relative pose and view synthesis tasks as proxies for analyzing the representation power of our learned embeddings. Nonetheless, our promising experimental results indicate our cross-domain embeddings may be useful for a variety of tasks.

2. Related Work

A number of recent works have introduced geometric structure to the feature representations of deep neural networks. The most common setting is to learn intermediate features that can be directly manipulated or transformed for a particular task. For example, in Rhodin et al. (2018), Worrall et al. (2017b), Cohen & Welling (2015), Hinton et al. (2011), Yang et al. (2015), and Kulkarni et al. (2015), geometric transformations can be directly applied to image features (in some cases disentangled pose features), in order to synthesize new views. In a related approach, Tatarchenko et al. (2016) uses an encoder-decoder architecture that augments pose information to the latent image embedding. One drawback of these methods is that they typically require full supervision, where both the geometric transformation parameters and the corresponding target image are available as supervision during training. Furthermore, training with source-target pairs requires covering a large sample space - synthesizing views from arbitrary relative 3D orientations requires sampling pairs from $\mathbf{SO}(3) \times \mathbf{SO}(3)$. In contrast, our model trains with a single image per example.

Different to all of the methods above, Homeomorphic VAEs (Falorsi et al., 2018) provide an unsupervised way to learn an $\mathbf{SO}(3)$-latent-embedding for images. However, it is presently unclear if this method can scale to practical scenarios (it requires a dense sampling of views to learn a continuous embedding, dealing with intra-class variations, etc).

While the examples above have all been applied towards the task of view synthesis, there also exist a variety of other approaches to this problem. Most relevant to our setting are the self-supervised methods that learn geometrically meaningful embeddings using differentiable rendering to match semantic maps (Yao et al., 2018), shading information (Henderson & Ferrari, 2018), fusing latent embeddings from multiple views, and improving synthesis using multiple rendering steps (Eslami et al., 2018).

Pose Estimation: The task of object pose estimation has been a long standing problem with numerous applications in computer vision and robotics. Most approaches can be categorized as keypoint-based or direct pose estimation as regression or classification. Keypoint-based methods for object pose estimation include Pavlakos et al. (2017) and Grabner et al. (2018), the former predicting semantic keypoints and the latter bounding box corners, from which object pose is determined from a PnP algorithm. Direct pose estimation methods include Tulsiani & Malik (2015) and Su et al. (2015) where classification is performed over a quantized viewpoint space. Kanezaki et al. (2018) train a joint 3D object classification and pose CNN from multiple views with unknown viewpoints, however the viewpoint sphere is sampled discretely providing limited resolution in the estimated pose. Mahendran et al. (2017) introduces a carefully designed CNN for viewpoint regression, analyzing different representations and geodesic loss functions, and Mousavian et al. (2017) introduce a MultiBin orientation regression network. KeypointNet (Suwajanakorn et al., 2018) learns category-specific semantic keypoints and their detectors using only a geometric loss. The 3D keypoints are also useful for determining relative pose, although the method struggles when exposed to arbitrary 3D rotations due to lack of rotation equivariance.

The key ingredient in our approach is a novel method to map 2D images to rotation-equivariant 3D shape embeddings, essentially encoding an image with 3D geometric structure. We note that the choice of geometric representation (spherical embeddings) is intentional in order to maintain rotation equivariance. Alternative geometric representations such as volumetric (e.g. single-view volumetric reconstruction from Tulsiani et al. (2017)) would not be rotation equivariant (although Weiler et al. (2018) could provide an alternative for certain tasks).

3. Method

In this section we detail our image embedding model. We begin by revisiting spherical CNNs (Section 3.1) as a means to learn rich equivariant embeddings for 3D shapes, and Section 3.2 introduces our cross-domain architecture that learns to map 2D images into the same embedding space. Sections 3.2.3 and 3.2.4 will describe how these image embeddings can be used for relative pose estimation and novel view synthesis.

3.1. Spherical CNNs

Spherical CNNs (Cohen et al., 2018; Esteves et al., 2018) produce 3D rotation equivariant feature maps for inputs defined on the sphere. In practice these methods have been useful for a variety of 3D shape analysis tasks as it is common for inputs to appear in arbitrary pose, for which equivariance is a particularly helpful property. In this work we adopt the spherical convolutional model introduced in Esteves et al. (2018) due to its efficiency and performance on 3D shape alignment tasks[1]. We briefly summarize the spherical CNN below (see Esteves et al. (2018) for details). For functions x_1, x_2 defined on the sphere, their convolution is defined as

$$(x_1 \star x_2)(p) = \int_{R \in \mathbf{SO}(3)} x_1(R\eta) x_2(R^{-1}p) dR, \tag{1}$$

where η is the north pole of the sphere (the stationary point under $\mathbf{SO}(2)$). This extends to K-channel inputs in a straightforward manner

$$(x_1 \star x_2)(p) = \sum_{k=0}^{K-1} \int_{R \in \mathbf{SO}(3)} x_{1,k}(R\eta) x_{2,k}(R^{-1}p) dR, \tag{2}$$

where $x_{\cdot,k}$ denotes the k-th channel.

These convolutions are the primary building blocks of spherical CNNs. We define s as a spherical CNN that maps K_{in}-channel spherical inputs to K_{out}-channel spherical feature maps. Precisely, in the single-channel case we have $s\colon L^2(S^2) \mapsto L^2(S^2)$ where $L^2(S^2)$ denotes square-integrability, which is necessary for the efficient evaluation of convolution in the spectral domain.

An important property of the spherical CNN is 3D rotation equivariance. For any $x\colon S^2 \mapsto \mathbb{R}^{K_{\text{in}}}$, we have

$$s(\Lambda_R x) = \Lambda_R s(x), \tag{3}$$

where Λ_R is the rotation operator by $R \in \mathbf{SO}(3)$[2]. Technically this equivariance is only approximate as the nonlinear activations (ReLU) and spatial pooling operations break the bandlimiting assumptions which otherwise guarantee equivariance. However, in practice these errors are negligible (Esteves et al., 2018).

To use spherical CNNs with 3D shapes, we must provide a map $r(M)$ which converts any 3D shape M to a spherical representation. While there are many choices for r we use the simple ray-casting technique of Esteves et al. (2018). Most importantly, $r(M)$ is equivariant to 3D rotations which ensures end-to-end equivariance of our 3D shape feature maps: $s(r(\Lambda_R M)) = \Lambda_R s(r(M))$.

3.2. Cross-domain spherical embeddings

In the previous section we summarized a rotation equivariant spherical CNN model for 3D shape inputs: $s(r(M))$. The primary objective of this work is to learn an *image* embedding that can capture similar underlying 3D shape

[1] It is important to note that we are tackling the more challenging problem of relative 3D pose from 2D images.

[2] We use Λ_R as a generic rotation operator that can be applied to 3D shapes and spherical functions, scalar or vector-valued. Interpretation should be clear from context.

properties and equivariant structure. Specifically, let us define an RGB image as the projection c of a shape M (c can be any usual camera projection model, e.g. perspective or orthographic). We seek a map $f(c(M))$ that captures the shape properties of M and retains an equivariant structure: $f(c(\Lambda_R M)) = \Lambda_R f(c(M))$. This is challenging because $c(M)$ is not 3D rotation equivariant as it is a camera projection, so we cannot have equivariance by construction. We propose to learn an approximately equivariant embedding model f using a spherical CNN for 3D shapes, i.e. a pretrained $s(r(M))$), as supervision: we wish to learn f such that $f(c(M)) = s(r(M))$. If learned successfully, the equivariance of f follows simply from (3):

$$\begin{aligned} f(c(\Lambda_R M)) &= s(r(\Lambda_R M)) \\ &= \Lambda_R s(r(M)) \\ &= \Lambda_R f(c(M)). \end{aligned} \tag{4}$$

Since $c(M)$ and $r(M)$ are fixed and not part of the trainable model, we substitute $y = c(M)$ and $x = r(M)$ going forward to simplify notation.

Learning f involves predicting high dimensional multi-channel spherical maps from a single image. The two major design challenges are deciding the structure of $f(y)$ and the training loss $\mathcal{L}(x, y)$ from predicted embedding $f(y)$ to the target ground truth $s(x)$. We describe first the training loss. For simplicity we describe the loss for a single channel (in general the loss is aggregated over the channels). The implementation of the spherical CNN represents the spherical function $s(x)$ on a grid via equirectangular projection. A discretized $s(x)$ of resolution $N \times N$ can be indexed by the pair $(\theta_i, \phi_j), i, j \in \{0, 1, ..., N-1\}$. The θ_i uniformly sample colatitude, and similarly ϕ_j uniformly sample azimuth. Since our target embeddings are unbounded, we found crucial to use a robust loss such as Huber[3], and a Huber breakpoint at 1 works well in practice. The loss follows, where $\mathcal{H}$ is the Huber loss, and a weight is introduced to account for the nonuniform equirectangular spherical sampling ($\sin(\theta)$ is proportional to the sample area):

$$\mathcal{L}(x, y) = \frac{1}{N^2} \sum_{i,j=0}^{N-1} \mathcal{H}(\sin(\theta_i)(f(y) - s(x))(\theta_i, \phi_j)) \tag{5}$$

$$\mathcal{H}(\alpha) = \begin{cases} 0.5\alpha^2 & \text{for } |\alpha| \leq 1, \\ |\alpha| - 0.5 & \text{otherwise.} \end{cases} \tag{6}$$

3.2.1. Architecture

We now describe the structure of our cross-domain embedding model $f(y)$. With $f(y)$ we are predicting spatially dense spherical feature maps from a single 2D image. Convolutional encoder-decoder architectures with skip connections such as U-Net (Ronneberger et al., 2015) or Stacked Hourglass (Newell et al., 2016) produce excellent results when some pixelwise association can be made between the input and output domains (e.g. for dense labeling tasks like semantic segmentation (Chen et al., 2018)). However, we must learn a cross-domain map from 2D image (Euclidean) to spherical functions. In this setting, architecture features such as skip connections are not only unnecessary but can be harmful by forcing the network to incorrectly consider associations across topologies.[4]

We consider an encoder-decoder architecture, with several rounds of downsampling from input image to a 1D vector, followed by rounds of upsampling from the 1D vector to the set of spherical feature maps. We follow the best practices for this kind of architecture proposed by Radford et al. (2015), employing a fully convolutional network with strided convolutions for downsampling and transposed convolutions for upsampling. We apply azimuthal circular padding after the 1D bottleneck, when the feature maps are expected to assume spherical topology. We also found performance improvements by replacing convolutional layers with residual layers (He et al., 2016). Figure 2 shows the architecture. See supplementary material for more details.

3.2.2. Target embeddings

The remaining decision is to select the appropriate target feature maps from $s(x)$. For all our experiments $s(x)$ is a 10 layer residual spherical CNN trained for ModelNet40 3D shape classification on 64×64 inputs (i.e. $r(M)$ produces a single-channel 64×64 output). The decision of which feature maps to use as the target is application-dependent. For category-based relative pose estimation, we want features that are void of instance level details, which is achieved by taking the target embedding from deeper layers. For view synthesis, we wish that the instance-level details are preserved, so we embed to an earlier layer. We employ the same pre-trained spherical CNN for all experiments (on ModelNet40, ObjectNet3D and ShapeNet), which shows generalization; more details in the supplementary material.

3.2.3. Relative pose estimation

The cross-domain embeddings produced by $f(y)$ are sufficient to recover the relative pose between pairs of images (even between different instances of the same object category). As $f(y)$ produces spherical feature maps that have been trained to be 3D rotation equivariant, we can apply 3D rotations directly to the feature maps. Relative orienta-

[3] median pose errors are ≈ 10 deg larger with L_1 or L_2

[4] Although cross-modal learning has been explored in different domains, e.g. Aytar et al. (2016), these methods predict representations in $\mathbb{R}^n$ from different modalities which is a simpler application of 1D and 2D CNNs.

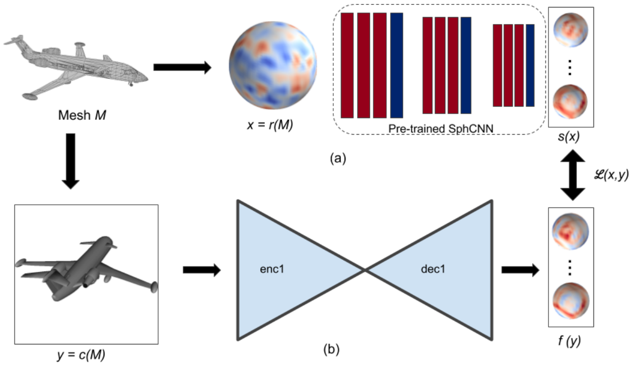

Figure 2. **Cross-domain spherical embeddings.** Given a 3D mesh, (a) we map it to a spherical function, and use a pre-trained spherical CNN to compute its spherical embedding. (b) During training, we render a view and learn the transformation to the target spherical embedding using an encoder-decoder. For inference, the inputs are 2D images and only the encoder-decoder part is used.

tion estimation is simply identifying the rotation that brings feature maps into alignment. For alignment we use a very simple cross-correlation measure. Given two images y_1, y_2 we estimate their relative pose as

$$\underset{R \in \mathbf{SO}(3)}{\arg\max}\, G(R) = \sum_{k=0}^{K-1} \int_{p \in S^2} f(y_1)_k(p) \cdot f(y_2)_k(R^T p) dp \tag{7}$$

Here the subscript k denotes the k-th spherical channel in the image embedding. $G(R)$ can be evaluated efficiently in the spectral domain (similar in spirit to spherical convolution, see Kostelec & Rockmore (2008); Makadia & Daniilidis (2010) for details and implementation).

The resolution of $G(R)$ depends on the resolution of the input spherical functions $f(y_1)$ and $f(y_2)$. Our learned feature maps have a spatial resolution of 16×16 which corresponds to a cell width of 22.5 deg at the equator, which we consider too coarse for precise relative pose. To increase resolution, we upsample the features by a factor of 4 using bicubic interpolation prior to evaluating (7).

This method can also be applied to estimate relative pose between an image y and mesh M, by computing the argmax correlation (7) between $f(y)$ and $s(r(M))$.

Recall, during training we take as input arbitrarily oriented meshes. A training example consists of only the target embeddings from the pretrained $s(r(M))$ and a single view rendered from a fixed camera $c(M)$. No orientation supervision is required, and the model never sees pairs of images together at training. This reduces the sample complexity and leads to faster convergence.

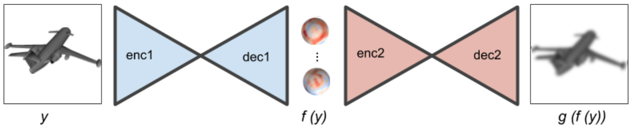

Figure 3. **Novel view synthesis training.** We learn the inverse map from spherical embeddings to 2D views. The map from 2D view to spherical embeddings (in blue) is the same as in Fig. 2 and is frozen during training. The synthesizer network (in red) reconstructs the same input view and is trained with an L_2 loss.

3.2.4. Novel view synthesis

The spherical embeddings learned by our method can also be applied towards novel view synthesis. The rotation equivariant spherical CNN feature maps undergo the same rotation as its inputs, so if we learn the inverse map that generates an image back from its embedding, we can rotate the embeddings and generate novel views.

We define the inverse map $g = f^{-1}$ such that $g(f(y)) = y$. If we let $y_1 = c(M)$ and $y_2 = c(\Lambda_R M)$ (i.e. $y_{1,2}$ are images of shape M before and after it undergoes a 3D rotation, respectively). It follows that

$$g(\Lambda_R f(y_1)) = g(f(y_2)) = y_2. \tag{8}$$

This gives us a way to generate a novel view of the 3D object under rotation, from the spherical embedding of a single view as follows (see Fig. 1 for illustration):

1. Obtain the embedding $f(y_1)$ of given view y_1,
2. Rotate the embedding by the desired $R \in \mathbf{SO}(3)$, obtaining $f(y_2) = \Lambda_R f(y_1)$,
3. Apply g to obtain the novel view $y_2 = g(f(y_2))$.

Since g is learning the inverse of f, we similarly design g as a convolutional encoder-decoder, which is trained from single views enforcing $g(f(y)) = y$ with a pixel-wise L_2 loss $\mathcal{L}_s(y) = \|g(f(y)) - y\|_2^2$ (see Fig. 3 for illustration).

4. Experiments

4.1. Datasets

We utilize the standardized large datasets of 3D shapes ModelNet40 (Wu et al., 2015) and ShapeNet (Chang et al., 2015) for most of our experiments.

Some methods must explicitly deal with the symmetries present in many shape categories (e.g. Saxena et al. (2009); Rad & Lepetit (2017)). Our method is immune to this problem by not requiring pose annotations. However, pose annotations are used for evaluation, therefore we limit our

			airplane			car			chair			sofa		
			med.	a@15	a@30	med.	a@15	a@30	med.	a@15	a@30	med.	a@15	a@30
2DOF	IB	Ours	**5.17**	**85.3**	**91.9**	**3.70**	**92.2**	92.5	**5.07**	**90.6**	**94.1**	**4.59**	**93.6**	**95.2**
		Regr.	16.9	46.3	68.7	6.55	83.5	**93.1**	13.7	53.9	78.3	17.3	43.2	69.4
		KpNet	6.95	79.4	91.5	div.	div.	div.	6.34	84.7	91.8	9.20	71.3	85.4
	CB	Ours	**6.24**	79.0	88.2	**4.73**	73.2	73.3	12.1	59.3	74.4	**10.8**	**58.7**	70.5
		Regr.	20.6	38.7	63.7	7.06	**82.4**	**92.5**	16.8	43.7	72.0	19.6	37.8	66.5
		KpNet	9.07	**79.4**	**91.5**	div.	div.	div.	**8.07**	**79.5**	**90.2**	15.1	49.8	**71.8**
3DOF	IB	Ours	**6.64**	**80.9**	**91.9**	**3.84**	**97.3**	**98.8**	**5.55**	**89.1**	**95.7**	**5.21**	**90.4**	**94.8**
		Regr.	45.4	12.6	31.3	9.83	69.0	86.5	21.7	31.3	64.3	22.2	34.8	61.4
		KpNet	14.9	50.3	76.6	9.12	70.4	80.9	10.8	66.7	85.3	25.0	27.4	57.3
	CB	Ours	**7.27**	**76.4**	**89.4**	**4.59**	**92.1**	**93.3**	**12.3**	**59.5**	77.3	**9.66**	**63.9**	**76.0**
		Regr.	44.4	14.1	32.1	10.5	66.5	85.6	25.6	25.1	57.2	24.5	30.9	58.1
		KpNet	16.3	46.0	75.0	10.7	64.4	77.6	13.6	55.4	**81.6**	37.4	12.7	39.8

Table 1. **ShapeNet relative pose estimation results.** We show median angular error in degrees (*med.*), accuracy *(*a@) at 15 and 30 deg for instance (*IB*) and category-based (*CB*), 2 and 3 degrees of freedom relative pose estimation from single views on ShapeNet. Comparison is against Mahendran et al. (2017) (*Regr.*) and Suwajanakorn et al. (2018) (*KpNet*). KeypointNet does not converge on the full 3DOF setting; we limit the viewpoints to a hemisphere when evaluating it. Note that we still outperform it.

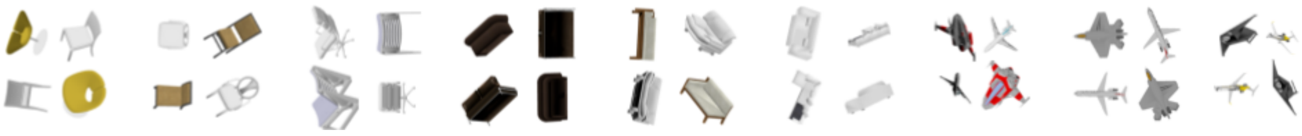

Figure 4. **Category-based relative pose estimation.** We render one object in the pose of the other using our estimated relative pose. *For each block, top:* Inputs 1 and 2, from the test set. *Bottom:* Mesh 2 rotated into pose 1, and mesh 1 rotated into pose 2. We render from the ground truth meshes for visualization purposes only; the inputs to our method are solely the 2D views and the output is the relative pose. Note how the alignment is possible even under large appearance variation.

experiments to categories which are largely free of symmetry and thus for which relative orientation is unique.

Symmetry is a problem for the evaluation of ShapeNet *airplanes*. Some of the instances (e.g. spaceships and flying wings) are fully symmetric around one axis, which results in non-injective embeddings and two possible correct alignments that differ by 180 deg. For meaningful evaluation we compute the errors up to symmetry for this class.

Recall that we are not estimating pose relative to a canonical object frame but rather relative object orientation from a pair of images. Thus, for training, we do not require that our dataset models come aligned per category, and in fact we introduce random rotations at training time. For evaluation, in order to quantify our inter-instance performance we require aligned shapes to determine the ground truth (see Section 4.2); for ModelNet40 we use the aligned version from Sedaghat & Brox (2015).

Multiple datasets have been proposed for object pose estimation, such as Pascal3D+ (Xiang et al., 2014), KITTI (Geiger et al., 2012), and Pix3D (Sun et al., 2018), but they do not exhibit large variation in viewpoints, especially camera elevation. For example Pascal3D+ has most elevations concentrated inside $[-10°, 10°]$ and the official evaluation only considers azimuthal accuracy. In our setting we explore geometric embeddings that can capture more challenging arbitrary viewpoints. Our results indicate that the problem of relative orientation from two views is quite difficult even for rendered images from ModelNet40 and ShapeNet. Our experiments with real images are limited to the *airplane* and *cars* categories of ObjectNet3D (Xiang et al., 2016), which have the largest variety of viewpoints among all categories.

4.2. Relative pose estimation

For training, we render views in arbitrary poses sampled from $\mathbf{SO}(3)$. We have two modes of evaluation: *instance* and *category* based. For category-based, we measure the relative pose error between each instance and 3 randomly sampled instances from the test set. For instance-based, we measure the error between each instance from the test set and 3 randomly rotated versions of itself. The error is the angle between the estimated and ground truth relative poses; given input ground truth poses R_1 and R_2 and estimated pose R, it is given by $\arccos\left((\mathrm{tr}(R_2^\top R_1 R) - 1)/2\right)$. We compare with the following methods.

Regression: We consider a method based on Mahendran et al. (2017), which treats pose estimation as regression. To keep the comparison fair we use approximately the same number of parameters in the encoder as in our networks. See supplementary material for more details. Mahendran et al. (2017) requires the ground truth pose with respect to a

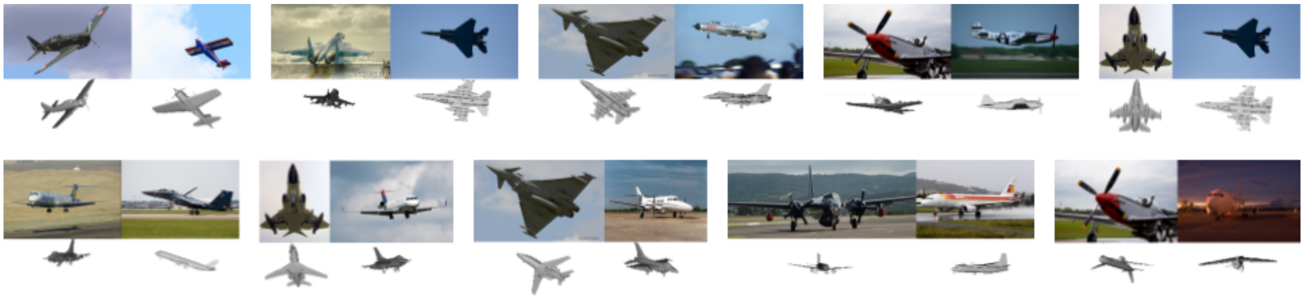

Figure 5. **Relative pose estimation for real images.** We render the mesh corresponding to one input in the pose of the other using our estimated relative pose. *For each block, top:* Inputs 1 and 2, from the test set. *Bottom:* Mesh 2 rotated into pose 1, and mesh 1 rotated into pose 2. Image pairs on the top row map to the same mesh in the dataset; on the bottom they map to different meshes. The bottom-right block shows a typical failure case due to symmetry. Meshes are used for visualization purposes only; the inputs to our method are the 2D images and the relative pose is estimated directly from their embeddings via cross-correlation (see text for details).

canonical orientation during training, whereas our method is self-supervised and can operate on unaligned meshes. We still outperform it even when allowing extra information, especially in the presence of 3DOF rotations.

KeypointNet: Suwajanakorn et al. (2018) introduce an unsupervised method of learning keypoints that can be used for pose estimation. Similarly to our method, it generates training data by rendering different views from meshes. It requires consistently oriented meshes for dominant direction supervision, whereas our method makes no assumptions about mesh orientation. While they show results for 2DOF rotations, only viewpoints on a hemisphere are considered, whereas we sample the whole sphere. We retrain and evaluate KeypointNet with full 2DOF and 3DOF rotations. Our method outperforms it on the more challenging scenarios.

Table 2 shows ShapeNet results. Figure 4 shows the 3DOF alignment quality on ShapeNet by rendering views using the estimated relative poses. We show results for ModelNet40 and for aligning meshes to images in the supplement.

4.2.1. Extension to real images

Most labeled real-world object pose estimation datasets have restricted pose variations. One dataset with sufficient variation of 3D poses is the airplane class in ObjectNet3D (Xiang et al., 2016). We assume object instance bounding boxes are provided (e.g. using an object detection network (Fathi et al., 2017)). We also experiment with the *cars* category by augmenting it with in-plane rotations to increase the pose variation. We train our model on image-mesh pairs and significantly outperform the method based on regression. *airplanes* numbers are up to a 180 deg rotation due to symmetry as explained in Section 4.4 (see bottom right of Figure 5 for an example). Table 2 shows the comparison while Figure 5 shows alignment results for *airplanes*.

		med err.	acc@15	acc@30
airplane	Ours	**13.75**	**53.40**	**76.60**
	Regression	36.52	16.70	40.40
car	Ours	**8.22**	**72.51**	**78.00**
	Regression	16.16	46.87	74.35

Table 2. Relative pose estimation results for real images from ObjectNet3D. We show median angle error in degrees and accuracy at 15 and 30 deg. We significantly outperform the regression method based on Mahendran et al. (2017).

4.3. Novel view synthesis

We evaluate novel view synthesis qualitatively[5]. Figure 6 shows the results for several generated views in different poses, with a single 2D image as input. We do not expect to generate realistic images here, since the embeddings do not capture color or texture and the generator is trained with a simple L_2 loss. Our goal is to show that the learned embeddings naturally capture the geometry, which is demonstrated by this example, where a simple 3D rotation of the spherical embeddings obtained from a single 2D image produces a novel view of the corresponding 3D object rotation. Adversarial and perceptual losses can be used in conjunction with our approach for refining the novel views (Karras et al., 2018; Wang et al., 2018). See supplementary material for further results from other categories.

4.4. Discussion

Our image to spherical cross-domain embeddings show quantitative improvements over state-of-art in relative 3D object pose estimation. Most existing literature shows results on a restricted set of rotations, and our numbers on 2DOF rotations are comparable to state-of-art. However,

[5] We attempted a method similar to Tatarchenko et al. (2016) as baseline, with and without adversarial losses, but results were poor for our large space of rotations.

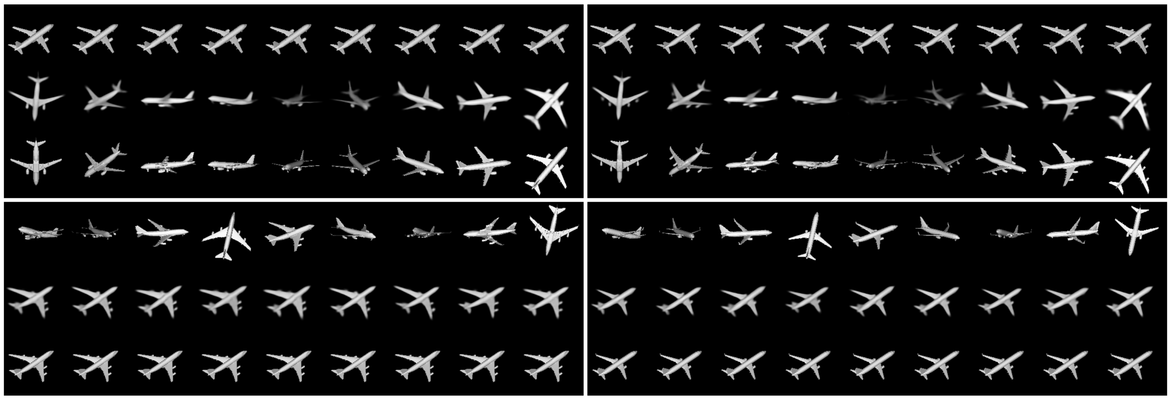

Figure 6. **Novel view synthesis.** Our embeddings are category based, capture both geometry and appearance, can be rotated as spheres, and can be inverted through another neural network. We can generate any new viewpoint from any given viewpoint. For each block: *top row:* inputs; *middle row:* novel views generated using our method; *bottom row:* ground truth views rendered from the original mesh. Top two blocks show different views generated from a single image; bottom two blocks show a single view generated from different images.

for full 3DOF rotations, relative pose estimation from 2D images is especially challenging for approaches which attempt to predict the pose directly from an image embedding, since it requires a combinatorially large training dataset. In contrast, our approach learns the mapping using fewer viewpoints and the corresponding spherical embeddings.

KeypointNet (Suwajanakorn et al., 2018) training failed to converge or converged to a bad model for cars 2DOF (noted 'div' in the table) and for the challenging 3DOF rotations. We found that KeypointNet converges if we limit the 3DOF setting to views on a hemisphere (instead of the full sphere). Our numbers for the full 3DOF space of rotations are still superior to KeypointNet's results for the limited 3DOF hemisphere.

Evaluation of the *airplane* class is problematic on ShapeNet due to the presence of symmetric instances (flying wings and some spaceships), which admit two possible alignments that differ by a 180 deg rotation. We also observe problems on ObjectNet3D, but in this case it's an approximate symmetry that sometimes is not captured by the low resolution spherical CNN feature maps. In both cases we consider the symmetry when evaluating the errors by making $\text{err}_{\text{sym}} = \min(\text{err}, \pi - \text{err})$. This metric is used for all methods on *airplanes*. Note that Suwajanakorn et al. (2018) also observe errors around 180 deg and benefit from this metric. ModelNet40 *airplanes* do not suffer from this issue.

Our method is capable of synthesizing any new viewpoint from any other given viewpoint for any instance of the class it was trained on. The categories with less appearance variation are easier to learn and produce sharper images. For all classes, however, we can verify that the full 3D information is captured by the embeddings.

5. Conclusion

In this paper, we explore the problem of learning expressive 3D rotation equivariant embeddings for 2D images. We proposed a novel cross-domain embedding that maps 2D images to spherical feature maps generated by spherical CNNs trained on 3D shape datasets. In this way, our cross-domain embeddings encode images with sufficient shape properties and an equivariant structure that together are directly useful for numerous tasks, including relative pose estimation and novel view synthesis.

We highlight two important areas for future work. First, the cross-domain embedding architecture is composed of a large encoder-decoder structure. The capacity of such a model is greater than what would be necessary for training traditional task-specific models (e.g. a relative pose regression network). This is due to the fact that we are solving a much harder problem: our model must learn a very expressive feature representation that can generalize to many applications. Nonetheless, in future work, we will explore ways to make this component more compact. Second, by construction, our work is tied to the spherical CNNs we use to supervise our embeddings. We will explore alternative rotation equivariant models supervise our training.

Going forward we will also try to improve our embedding representation so that they can be useful for even more challenging tasks such as textured view synthesis for example.

6. Acknowledgements

K. Daniilidis is grateful for support through these grants: NSF-IIP-1439681 (I/UCRC), NSF-IIS-1703319, NSF MRI 1626008, ARL RCTA W911NF-10-2-0016, ONR N00014-17-1-2093, ARL DCIST CRA W911NF-17-2-0181, the DARPA-SRC C-BRIC, and by Honda Research Institute.

References

Aytar, Y., Vondrick, C., and Torralba, A. Soundnet: Learning sound representations from unlabeled video. In *Advances in Neural Information Processing Systems*, 2016.

Chang, A. X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., and Yu, F. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015.

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., and Adam, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *European Conference on Computer Vision (ECCV)*, 2018.

Cohen, T. S. and Welling, M. Transformation properties of learned visual representations. In *International Conference on Learning Representations (ICLR)*, 2015.

Cohen, T. S. and Welling, M. Group equivariant convolutional networks. In *International Conference on Machine Learning, ICML*, 2016.

Cohen, T. S., Geiger, M., Köhler, J., and Welling, M. Spherical CNNs. In *International Conference on Learning Representations (ICLR)*, 2018.

Eslami, S. M. A., Jimenez Rezende, D., Besse, F., Viola, F., Morcos, A. S., Garnelo, M., Ruderman, A., Rusu, A. A., Danihelka, I., Gregor, K., Reichert, D. P., Buesing, L., Weber, T., Vinyals, O., Rosenbaum, D., Rabinowitz, N., King, H., Hillier, C., Botvinick, M., Wierstra, D., Kavukcuoglu, K., and Hassabis, D. Neural scene representation and rendering. *Science*, 360(6394):1204–1210, 2018.

Esteves, C., Allen-Blanchette, C., Makadia, A., and Daniilidis, K. Learning SO(3) equivariant representations with spherical cnns. In *The European Conference on Computer Vision (ECCV)*, September 2018.

Falorsi, L., de Haan, P., Davidson, T. R., Cao, N. D., Weiler, M., Forré, P., and Cohen, T. S. Explorations in homeomorphic variational auto-encoding. *CoRR*, abs/1807.04689, 2018.

Fathi, A., Korattikara, A., Sun, C., Fischer, I., Huang, J., Murphy, K., Zhu, M., Guadarrama, S., Rathod, V., Song, Y., and Wojna, Z. Speed and accuracy trade-offs for modern convolutional object detectors. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Geiger, A., Lenz, P., and Urtasun, R. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

Grabner, A., Roth, P. M., and Lepetit, V. 3d pose estimation and 3d model retrieval for objects in the wild. *CoRR*, 2018.

He, K., Zhang, X., Ren, S., and Sun, J. Identity mappings in deep residual networks. *CoRR*, abs/1603.05027, 2016.

Henderson, P. and Ferrari, V. Learning to generate and reconstruct 3d meshes with only 2d supervision, 2018.

Hinton, G. E., Krizhevsky, A., and Wang, S. D. Transforming auto-encoders. In *Proceedings of the 21st International Conference on Artificial Neural Networks (ICANN)*, pp. 44–51, 2011.

Jaderberg, M., Simonyan, K., Zisserman, A., and kavukcuoglu, k. Spatial transformer networks. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 2017–2025, 2015.

Kanezaki, A., Matsushita, Y., and Nishida, Y. Rotationnet: Joint object categorization and pose estimation using multiviews from unsupervised viewpoints. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Karras, T., Aila, T., Laine, S., and Lehtinen, J. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018.

Kostelec, P. J. and Rockmore, D. N. Ffts on the rotation group. *Journal of Fourier Analysis and Applications*, 14 (2):145–179, 2008.

Kulkarni, T. D., Whitney, W. F., Kohli, P., and Tenenbaum, J. Deep convolutional inverse graphics network. In *Advances in Neural Information Processing Systems (NIPS)*, 2015.

Mahendran, S., Ali, H., and Vidal, R. 3d pose regression using convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017.

Mahjourian, R., Wicke, M., and Angelova, A. Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Makadia, A. and Daniilidis, K. Spherical correlation of visual representations for 3d model retrieval. *International Journal of Computer Vision (IJCV)*, 89(2):193–210, 2010.

Mousavian, A., Anguelov, D., Flynn, J., and Kosecka, J. 3d bounding box estimation using deep learning and geometry. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Newell, A., Yang, K., and Deng, J. Stacked hourglass networks for human pose estimation. In *European Conference on Computer Vision (ECCV)*, pp. 483–499. Springer, 2016.

Pavlakos, G., Zhou, X., Chan, A., Derpanis, K. G., and Daniilidis, K. 6-DoF object pose from semantic keypoints. In *International Conference on Robotics and Automation (ICRA)*, 2017.

Rad, M. and Lepetit, V. BB8: A scalable, accurate, robust to partial occlusion method for predicting the 3d poses of challenging objects without using depth. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

Radford, A., Metz, L., and Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, abs/1511.06434, 2015.

Rhodin, H., Salzmann, M., and Fua, P. Unsupervised geometry-aware representation for 3d human pose estimation. In *The European Conference on Computer Vision (ECCV)*, September 2018.

Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)*, 2015.

Saxena, A., Driemeyer, J., and Ng, A. Y. Learning 3-d object orientation from images. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2009.

Sedaghat, N. and Brox, T. Unsupervised generation of a viewpoint annotated car dataset from videos. In *IEEE International Conference on Computer Vision (ICCV)*, 2015.

Su, H., Qi, C. R., Li, Y., and Guibas, L. J. Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3d model views. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

Sun, X., Wu, J., Zhang, X., Zhang, Z., Zhang, C., Xue, T., Tenenbaum, J. B., and Freeman, W. T. Pix3d: Dataset and methods for single-image 3d shape modeling. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Suwajanakorn, S., Snavely, N., Tompson, J. J., and Norouzi, M. Discovery of latent 3d keypoints via end-to-end geometric reasoning. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 2063–2074, 2018.

Tatarchenko, M., Dosovitskiy, A., and Brox, T. Multi-view 3d models from single images with a convolutional network. In *The European Conference on Computer Vision (ECCV)*, pp. 322–337, 2016.

Tulsiani, S. and Malik, J. Viewpoints and keypoints. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

Tulsiani, S., Zhou, T., Efros, A. A., and Malik, J. Multi-view supervision for single-view reconstruction via differentiable ray consistency. In *IEEE Conference on Computer Vision and Pattern Regognition (CVPR)*, 2017.

Wang, T.-C., Liu, M.-Y., Zhu, J.-Y., Tao, A., Kautz, J., and Catanzaro, B. High-resolution image synthesis and semantic manipulation with conditional gans. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Weiler, M., Geiger, M., Welling, M., Boomsma, W., and Cohen, T. 3d steerable cnns: Learning rotationally equivariant features in volumetric data. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*. 2018.

Worrall, D. E., Garbin, S. J., Turmukhambetov, D., and Brostow, G. J. Harmonic networks: Deep translation and rotation equivariance. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, 2017a.

Worrall, D. E., Garbin, S. J., Turmukhambetov, D., and Brostow, G. J. Interpretable transformations with encoder-decoder networks. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017b.

Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., and Xiao, J. 3d shapenets: A deep representation for volumetric shapes. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2015.

Xiang, Y., Mottaghi, R., and Savarese, S. Beyond pascal: A benchmark for 3d object detection in the wild. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2014.

Xiang, Y., Kim, W., Chen, W., Ji, J., Choy, C., Su, H., Mottaghi, R., Guibas, L., and Savarese, S. Objectnet3d: A large scale database for 3d object recognition. In *European Conference Computer Vision (ECCV)*, 2016.

Yan, X., Yang, J., Yumer, E., Guo, Y., and Lee, H. Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

Yang, J., Reed, S. E., Yang, M.-H., and Lee, H. Weakly-supervised disentangling with recurrent transformations for 3d view synthesis. In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems (NIPS)*, pp. 1099–1107. Curran Associates, Inc., 2015.

Yao, S., Hsu, T. M. H., Zhu, J.-Y., Wu, J., Torralba, A., Freeman, B., and Tenenbaum, J. 3d-aware scene manipulation via inverse graphics. *arXiv preprint arXiv:1808.09351*, 2018.