View Generalization for Single Image Textured 3D Models

Anand Bhattad[1*]   Aysegul Dundar[2,3]   Guilin Liu[3]   Andrew Tao[3]   Bryan Catanzaro[3]

[1]University of Illinois at Urbana-Champaign   [2]Bilkent University   [3]NVIDIA
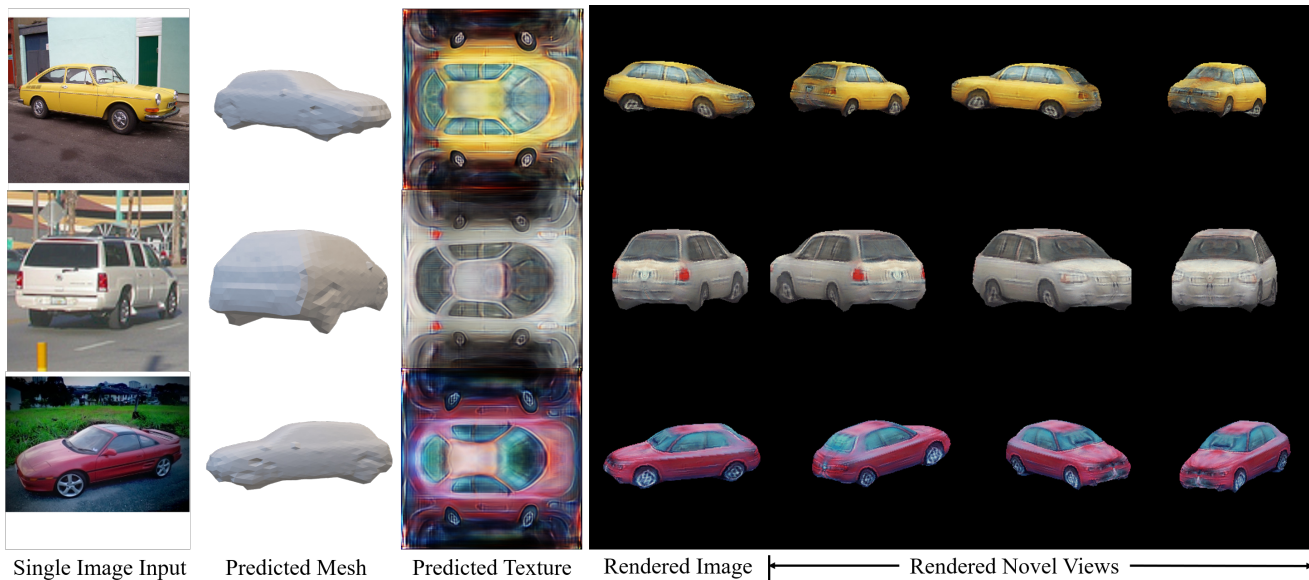
Figure 1: Given a single 2D image input, we infer high quality textured 3D models – a triangular mesh and a texture map, which can then be used to render novel views. Our inferred models compare well with the source image when rendered from that view. More important, our method displays good *view generalization* – new views of an inferred model look like real pictures of that object.

Abstract

Humans can easily infer the underlying 3D geometry and texture of an object only from a single 2D image. Current computer vision methods can do this, too, but suffer from view generalization *problems – the models inferred tend to make poor predictions of appearance in novel views. As for generalization problems in machine learning, the difficulty is balancing single-view accuracy (cf. training error; bias) with novel view accuracy (cf. test error; variance). We describe a class of models whose geometric rigidity is easily controlled to manage this tradeoff. We describe a cycle consistency loss that improves view generalization (roughly, a model from a generated view should predict the original view well). View generalization of textures requires that models share texture information, so a car seen from the back still has headlights because other cars have headlights. We describe a cycle consistency loss that encourages model textures to be aligned, so as to encourage sharing. We compare our method against the state-of-the-art method and show both qualitative and quantitative improvements.*

1. Introduction

People can easily infer the 3D geometry and texture of an object from a single 2D image. Likely, we can reason about the shape and the appearance of an object seen from only a single view because we have seen many objects from many views. Experience enables us to infer a reasonable estimate of shape and appearance of a novel object from a familiar category, even though there is within category variation. Current algorithms can recover 3D shape and texture from a single image, but suffer from severe *view generalization* problems: the recovered representation, viewed in a new camera, tends to look unrealistic. Either the inferred 3D mesh is squashed when viewed from other viewpoints [16, 4] or the inferred texture are muddled [16], and also not sharp enough with missing details [4]. For a faithful re-
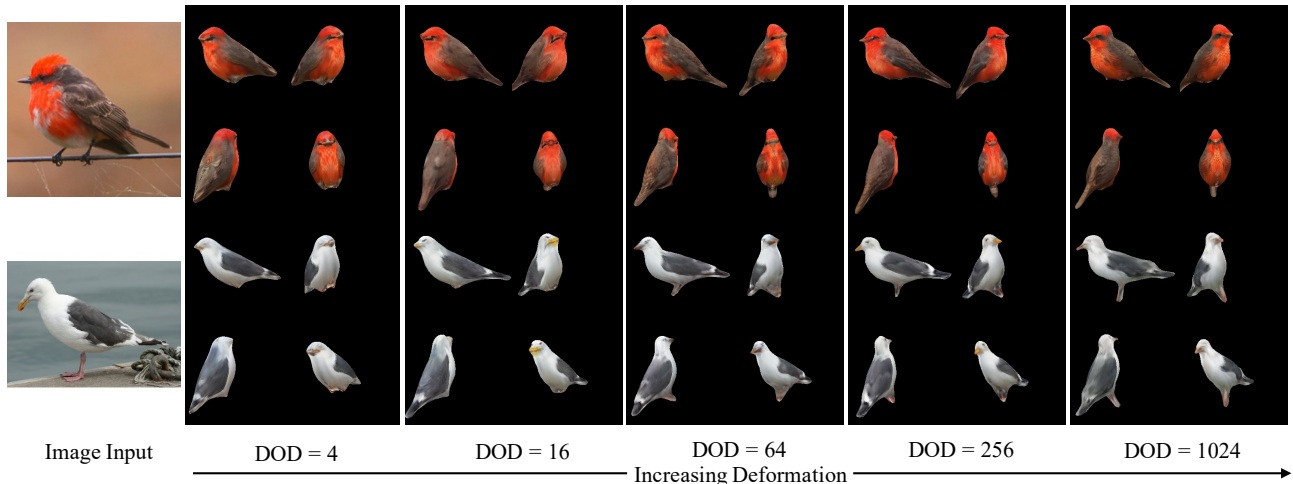

Figure 2: **Controllable Deformation.** Our approach is built upon a 2D convolutional deformation where one can easy control the extent of deformation by changing degrees of deformation (DOD) or spatial resolution of our predicted 2D deformation map. A higher DOD generally yields a more flexible model (eg the beak of the gull), with better fit to the source view but the possibility of view generalization problems (eg gull's eye in 1024). We report results for a fixed DOD, but note that specific instances might perform better at different DOD's – an artist might choose the best DOD for a particular source image.

covery of 3D models, current state-of-the-art (SOTA) methods rely on a finely curated multi-view datasets or use of synthetic images [3, 6] which do not translate well to real image domains. Therefore, in this work, we focus on inferring realistic textured 3D models – meshes and texture maps; from a collection of single-view real images and by using only available 2D supervision. We not only synthesize superior results from the original image view but also our synthesis generalizes across views with a coherent geometry and convincing texture prediction (Fig.1).

Good view generalization requires a geometric deformation model that can balance bias (source view) and variance (novel views). A highly flexible model may fit the original image well, but look bad from other directions; too rigid a model may generalize moderately well, but not fit the original image. Furthermore, good view generalization requires texture models that can share details across instances as appropriate: the way to know that this car has (say) headlights, even though they are not visible in the original view, is to know that all cars have headlights, and borrow headlights from some other model.

Our geometric model uses a controllable convolutional deformation approach that predicts a single UV deformation map over a spherical surface. This allows us to control the extent of deformation by varying the predicted spatial map resolution. The convolutional deformation is previously proposed by [32], here we use it for controllability by changing the degrees of deformation 3(Fig. 2). Our texture model is, again, on a single UV map, allowing us to use cycle consistency losses to share texture features. Our geometry and texture models are controlled with two consistency losses: i) Rotation GAN cycle consistency to deal with deformation from canonical views, encouraging their appearance and shape to be realistic as the original view images. ii) Texture mesh alignment consistency loss that shares appearance information across images providing strong spatial cues for inferring occluded texture.

Our experiments show that our framework built upon these philosophies and principles strongly improves the overall fidelity of synthesized textured 3D models (Fig. 1). We show that our framework allows us to control the flexibility of our deformation that is specifically useful for deforming non-rigid object categories like birds where different images undergo different degrees of deformation (Fig. 2). We also show that leveraging a few handfuls of multiple mesh templates instead of a single mesh template can improve overall recovered 3D inference quality without requiring the need of finely curated multi-view dataset.

Current SOTA methods are evaluated on intersection over union (IoU) compared to the ground truth mask from the original view. But a good mIoU from the original view may generate meshes that look poor from other views. The evaluation misses the point of 3D models - they are intended to predict appearance from *new* viewpoints. We evaluate mIoU for comparison, but evaluate view generalization by comparing multiple renderings from novel viewpoints with real images of real objects (using Fréchet Inception Distance (FID) scores [12]). We also conduct user studies comparing our inferred 3D models with the baseline methods. In summary, our main contributions are:

- A controllable convolutional deformation approach for better recovery of 3D geometry of the objects.
- Two novel cycle consistency losses that strongly improves overall inferred textured 3D models.
- High-fidelity textured 3D model synthesis from a single image both qualitatively and quantitatively.

2. Related Work

Generative Adversarial Networks (GANs). Recently GAN [10] based models have achieved realistic image synthesis on various objects [18, 43, 19]. However, for this technology to be used in gaming, simulators, and virtual applications, they need to be fully controllable. Recent work controls image generation by conditioning output on different type of inputs such as natural and synthetic images [17, 48, 23, 44, 7, 28], landmarks [27, 39, 8], and semantic maps [14, 40, 31]. Among these methods, [27, 8, 15] disentangle images into pose and appearance in an unsupervised way, and show control over pose during inference. However, since they do not disentangle images into their 3D attributes, the control and manipulation over generated images are still limited. Other models [18, 43, 19] are shown to learn an implicit 3D knowledge of objects without 3D supervision and the viewpoint of the synthesized object can be manipulated by latent codes. InfoGAN [5] and β-VAE [13] are proposed to learn interpretable disentangled features in an unsupervised manner. However, there is no clear factorization between 3D object models and object appearance. HoloGAN [30] for more explicit disentanglement, additionally incorporates a 3D rigid-body transformation module and a 3D-to-2D projection module in the network, and learns 3D voxel feature representations of the world. It provides the ability to control the pose of the generated objects better, however, still lacks explicit physical 3D model.

Single Image Textured 3D Inference. Differentiable renderers [26, 20, 24, 4, 33] enable inferring 3D properties from 2D images via gradient based optimization [16, 9, 22, 11]. However, inferring 3D shape, texture, camera parameters, and lighting from single 2D images is inherently ill-posed. To circumvent this problem, multiview image datasets have been used especially on synthetic datasets [3, 37, 36]. Multi-view images are not available for many datasets, and learning multi-view consistent shape and texture predictions from single image datasets is an important research area. In that spirit, [32] is similar to our work where GAN training is utilized to discriminate invisible and visible patches from mesh and texture predictions, enforcing the network to generate consistent predictions among visible and invisible areas. In [32], a two step training pipeline is proposed; first texture and mesh networks are trained via a reconstruction loss to predict visible regions and second these networks are trained from scratch with GAN loss such that invisible and visible areas look unrecognizable. The final networks do not encode 3D attributes from images but instead generate them from Gaussian noise which makes it not comparable to ours.

Cycle Consistency. Cycle Consistency enforces that transforms along the cycles are close to the identity function. It has been used on various computer vision tasks from image translation [48, 1], video interpolation [34, 25] to op-

tical flow estimation [29] just to name a few. Similar to our work, cycle consistency has been extensively used to disentangle image attributes such as pose and appearance disentanglement for person re-identification problem [47, 45], for cross identity motion transfer for arbitrary objects [15], shape and pose disentanglement for 3D meshes [47], and learn textures from 2.5D skecteches [49]. A rotation consistency [46] loss has been used to train pix2pixHD [40], a 2D image-to-image translation method for rotating faces. That work takes advantage of 3D face modeling by using off-the shelf 3D-fitting network [50], and renders images in novel views with missing textures. Pix2pixHD is trained to inpaint missing parts in the images. In our work, we use cycle consistency among rotations of the same object and texture mesh alignment between two different objects when training encoders for 3D geomerty and texture predictions.

3. Approach

First, we start with introducing the overall pipeline as given in Fig. 3, and then introduce the cycle consistency losses which provide a strong supervision signal for parts that are not visible from the original view.

3.1. Controllable Convolutional Deformation

We follow a similar encoder-decoder architecture from Chen *et al*. [4] that encodes images into 3D geometry and decodes textures, and then both are fed into a differentiable renderer to reconstruct input images with image-view camera parameters as shown in Fig. 3. We use DIB-R [4] as our differentiable renderer.

The key to infer strong 3D geometry is a detailed local control of model complexity and mesh deformation. This is crucial because we want the model to be flexible enough for every image and rigid enough for realistic rendering of novel views. Previous methods [16, 4] use a fully connected linear layer to predict deformation of a template mesh. This strategy has several disadvantages. First, each vertex is independently deformed. Second, there is no controllability over deformation when using a linear layer; that is if they need to be large (flexible) or small (rigid). Specially, for non-rigid objects like birds, one needs to be sufficiently flexible about the extent of deformation. Finally, these deformations are bound to a specific set of vertices of a template mesh and are difficult to re-sample for any new vertices (say learn deformation for low-poly mesh but synthesis a high-poly mesh at inference).

In contrast to previous methods, we use a spherical mapping to map our surface, as represented by a mesh to a sphere [32]. Then our deformation is a representation on the function of sphere directly with a fixed surface topology. We predict a single spatial deformation UV map for this spherical representation using a convolutional architecture as shown in Fig. 3. We then sample deformation for
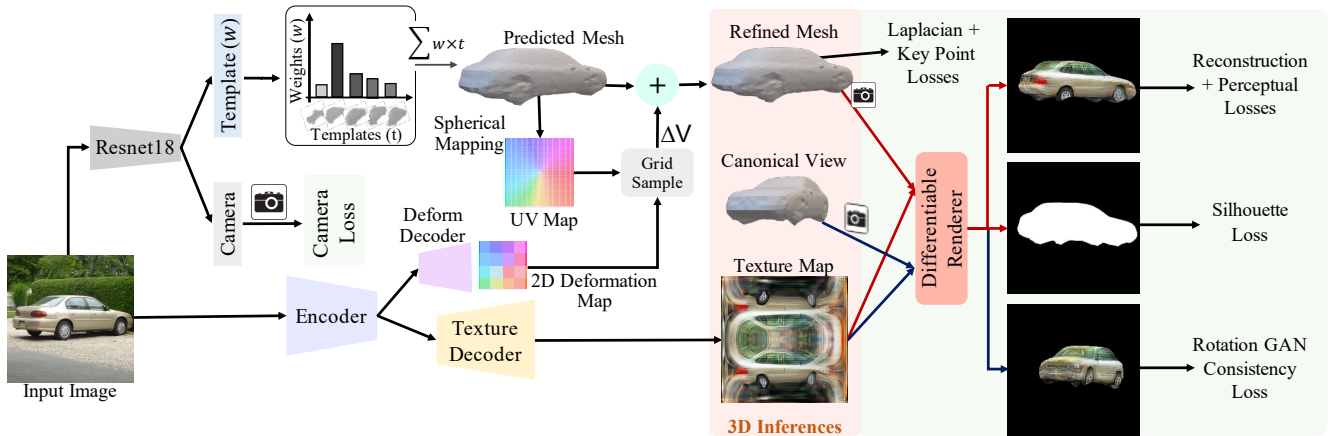
Figure 3: **Architecture overview.** From a single input image, we learn to infer object's 3D geometry (a mesh) and its texture (an UV map). Our final geometry is learned by a weighted combination of object templates and refined by applying adequate deformation to it. For the deformation, we follow a convolutional approach over a spherical representation mapped as an UV map and sample final deformations of our predicted mesh template from this deformation map. We also learn camera parameters with provided ground-truth, but during training we render the prediction with a known camera for the stability. We use various losses that are based on 2D; an image loss between input image and the rendered prediction, a silhouette loss between the object mask and prediction, and various other regularization and consistency losses described in Section 3.

each of the template's vertex position from the predicted UV deformation map. We show that this deformation allows us to better control deformation complexity as can be seen from Fig. 2. For large deformation models, we predict a high-resolution spatial deformation map (32×32) allowing models to be flexible enough. For small deformation, we predict a small-resolution deformation map (2×2 or 4×4) having a stronger constraint over deformation as we deform only limited positions of the sphere. We refer these controllability as degree of deformation (DOD). For eg., $DOD = 4$ means our 2D deformation is of 2×2 spatial resolution. Since, the learned deformation is a function of sphere and not the template we could then also easily sample deformation for as many vertices without crippling learning and computational efficiency. We leverage this and infer 3D meshes with over 2500 vertices and 6000 faces and are about $4\times$ more dense than previous methods.

Mesh templates. In our overall architecture, we follow a different approach than previous methods when learning geometry. We take advantage of provided 3D mesh templates from datasets if available. Previous methods [16, 4] start with mean templates and learn to deform it. We instead use multiple templates provided by PASCAL3D+ dataset [42], and a mean template for CUB dataset [41]. The network learns w, the template weight, to choose one from n templates it needs to start with for an image. However, we do not have any supervision for these templates, so we predict normalized weights (w) of each template (t) and take their weighted sum as our predicted mesh template. We also use a learnable scale-factor (s) for each of these templates

to adapt our templates to appropriate size. We then refine this predicted mesh by sampling deformation of all vertices (ΔV) using the convolutional UV deformation map. Our final refined mesh vertices are $V = \sum_{i=1}^{n} s_i \times w_i \times V_i^t + \Delta V$.

We show that making use of only a few additional templates improves our textured 3D inference significantly without requiring multi-view dataset. For pre-processing of mesh templates, we use same procedure described in [21]. We use all meshes that are provided in PASCAL3D+ dataset and ignore those templates that fails during the following preprocessing steps. We first remove all interior triangles and faces by simplifying mesh into voxels. Next, we convert these voxels into a mesh by running a marching cubes algorithm. We then further simplify mesh by closing holes and decimating mesh. Finally, we use spherical parameterization and map sphere to our simplified mesh.

Losses. Our base losses are same as DIB-R [4] and use DIB-R as our strong baseline. We use $\mathcal{L}_1$ image reconstruction loss between the input image (I) and the rendered image (I_r) and also perceptual losses from Alexnet (Φ) at different feature layers (j) between these images.

$$\mathcal{L}_{recon} = ||I - I_r||_1 \qquad (1)$$

$$\mathcal{L}_{percp} = ||\Phi_j(I) - \Phi_j(I_r)||_2 \qquad (2)$$

For shapes, we use an IoU loss between the silhouette rendered (S_r) and the silhouette (S) of the input object.

$$\mathcal{L}_{sil} = 1 - \frac{||S \odot S_r||_1}{||S + S_r - S \odot S_r||} \qquad (3)$$

Similar to [4, 24], we also regularize predicted mesh using a smoothness loss and laplacian loss ($\mathcal{L}_{lap}$) constraining

neighboring mesh triangles to have similar normals. Chen et al [4] uses camera as latent by not learning them and using the ground truth cameras. Similar to [16], we learn to predict camera pose using a simple $\mathcal{L}_2$ regression loss ($\mathcal{L}_{cam}$) between ground truth and predictions that we get from Resnet18 backbone as shown in our Fig. 3. We also predict keypoints from the inferred mesh and apply $\mathcal{L}_2$ regression loss comparing to the ground truth key point annotations ($\mathcal{L}_{KP}$). Finally, we regularize deformation to be small and hence minimize overall deformation to be as close to zero ($\mathcal{L}_{deform} = ||\Delta V||$). Following are our base losses:

$$\mathcal{L}_{baseline} = \lambda_r \mathcal{L}_{recon} + \lambda_p \mathcal{L}_{percp} + \lambda_s \mathcal{L}_{sil} \\ + \lambda_c \mathcal{L}_{cam} + \lambda_{kp} \mathcal{L}_{KP} + \lambda_d \mathcal{L}_{deform} + \lambda_{lap} \mathcal{L}_{lap} \quad (4)$$

Our λ coefficients are: $\lambda_r = 20$, $\lambda_p = 0.5$, $\lambda_s = 5.0$, $\lambda_c = 1$, $\lambda_{kp} = 50.0$, $\lambda_d = 2.5$, $\lambda_{lap} = 5.0$.

3.2. Cycle-Consistency Losses

We want our learning problem to be set in such a way that both texture and geometry information is shared across image collections efficiently while training and hence they can be inferred accurately during run-time. Our use of multiple templates largely improves over inferring geometry without extreme deformation of mesh triangles for views other than what that can be observed. We also benefit significantly by introducing two novel cycle-consistency losses such that both texture and geometry information are coherently shared across image collections.

Our first loss is a GAN based rotation cycle consistency loss. We use GAN losses on rendered images from novels views to be synthesized as real along with implicitly imposing a multi-view cycle consistency constraint (Fig. 4). Our second loss is a texture-mesh alignment cycle consistency loss. We use this to align texture to meshes irrespective of their views and shapes (Fig. 5) and share information whenever possible. For example, the predicted headlights in texture map from one car image should be near about the same place when predicted from another car image.

3.2.1 Rotation GAN Cycle Consistency

Single image 3D inference is an ill-posed problem. There is no supervision from other views and hence gradients to update geometry and texture. [38, 24] control these effects by imposing a multi-view consistency loss and renders multiple views given a single image on synthetic generated Shapenet dataset [3]. However, this approach is not scalable for real images as it is hard to collect multi-view images.

We impose multi-view consistency implicitly without requiring image-collections from multiple views. The cycle based multi-view consistency is imposed as follows: say M is a function which outputs 3D meshes given an input image, T is a texture network, and R is a differentiable renderer
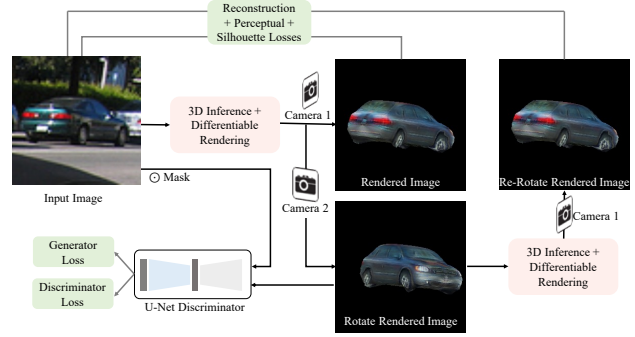

Figure 4: **Rotation GAN Cycle Consistency**. We extract geometry and texture predictions from input images, and render them from novel views. From those rendered images, we again predict geometry and texture but this time render them with the original camera parameters. The final images should match the input images. The rendered novel views must also appear realistic to a discriminator and is also trained with a generative loss.

function that generates the image. Given an image X_1 and its corresponding camera pose C_1, we first infer this image into its geometric and texture representations, and render an intermediate image from a novel camera view, C_2.

$$I_1 = \mathbf{R}(\mathbf{M}(X_1), \mathbf{T}(X_1), C_2)$$

Using this intermediate image, we infer back the geometry and texture representations, and render back an image with the original view camera pose, C_1.

$$X_1' = \mathbf{R}(\mathbf{M}(I_1), \mathbf{T}(I_1), C_1)$$

We apply a cycle consistency check between X_1 and X_1' using the reconstruction loss (Eq 1) and perceptual loss (Eq 2). Additionally, we train a U-Net discriminator [35] that can distinguish if the image is real or fake at global level as well as individual pixels level. For fake images, we use the novel view images, I_1, which provides gradient signals from the intermediate image. These losses implicitly impose multi-view consistency; GAN loss penalizes if rendered novel views are not realistic for both the shape and appearance, and the cycle consistency loss regularizes the network to infer predictions for occluded regions consistent with the original view. This constraint significantly improves our overall 3D synthesis.

3.2.2 Texture Mesh Alignment Cycle Consistency

Our goal is to not only synthesize accurate 3D shapes but also realistic texture maps. However, since we train our network only with single-view images, the network predicts perfect texture and shape only for the same camera view, ignoring alignment of shapes and texture for novel views resulting in visible inconsistencies. We can view this as overfitting (biased) to the original view. Key to better novel view texture synthesis is avoiding such overfitting by building a
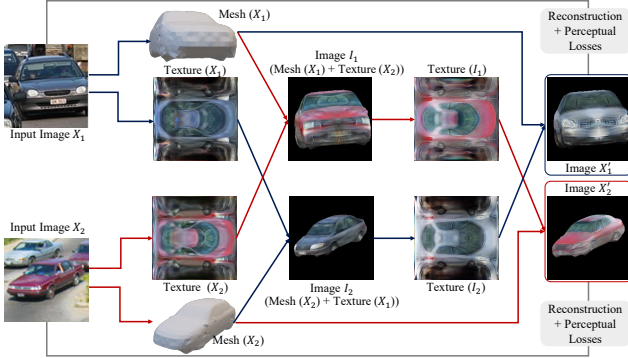
Figure 5: **Texture Mesh Alignment Cycle Consistency** enables texture representations to share when possible. We extract geometry and texture from randomly paired images and render novel images with swapped textures between these samples, then swap again and compare to original images. This encourages the texture representation to store details like (say) headlights in the same places, and so facilitates sharing textures across models.

robust prior over appearance, enabling sharing information between images to recover realistic texture for positions that are not visible from original view. We achieve this by texture mesh alignment cycle consistency (Fig. 5). Given two images, X_1 and X_2, and their corresponding camera poses C_1 and C_2, we infer their 3D mesh and textures. We then swap textures to reconstruct two new intermediate images:

$$I_1 = R(M(X_1), T(X_2), C_1)$$
$$I_2 = R(M(X_2), T(X_1), C_2)$$

We now infer texture from these two new images and swap textures back again to reconstruct the original images:

$$X_1^{'} = R(M(X_1), T(I_2), C_1)$$
$$X_2^{'} = R(M(X_2), T(I_1), C_2)$$

The final renderings should match the input images since we swap textures back-and-forth. The intuition behind this cycle constraint is to regularize the network from overspecializing to each example view (reduce bias). It also provides spatial cues for how the texture should behave to novel views and occluded areas by extracting representations from sampled images during back-and-forth swapping. This strongly helps in recovering accurate and realistic textures across novel views (see last column in Fig. 6).

4. Experiments

In this section we describe our experimental evaluations.
Datasets. We primarily use Pascal 3D+ dataset [42] - Car category for our evaluations. We also evaluate our method for birds using CUBs [41] dataset. We use same train-test split as that provided by DIB-R [4]. We use 7 templates provided by Pascal3D+ dataset as our starting templates for cars after post-processing as described in our Section 3. For birds, we start with only a single template.

Baselines and Ablations. For our baselines, we mainly compare our method against the previous SOTA, DIB-R [4]. We use their provided pretrained models for comparison. For DIB-R baseline, note that we used ground truth camera poses as they do not learn to infer camera poses. For all our evaluations, we use inferred camera poses. Another change from DIB-R is the use of reflection symmetry for both predicting geometry and texture maps. DIB-R does not use symmetry for geometry as imposing this constraint hurts their overall quality of inferences. We also perform ablation studies to verify the extent of improvements over using multiple templates and single templates. The effectiveness of convolution deformation over linear layer deformation and also the role of our two cycle consistency losses.

Model Details. We show our model pipeline in Fig. 3. We have two feature extractors: one from Resnet18 that is used to predict camera poses and template weights (when using over one template) and second for predicting deformation and texture maps. We predict camera poses and template weights using a single linear layer over Resnet18 features. Our predicted mesh shape then for an image is a simple weighted average over template vertices which we refine later using the deformation predicted. For predicting deformation and texture maps, we use a U-Net architecture. Our texture maps predicted are of 256×256 resolution. For our evaluations, we use deformation with $DOD = 4$ or 2×2 spatial resolution for cars and $DOD = 256$ or 16×16 spatial resolution for birds. We sample deformation using a bi-linear grid interpolation for our predicted mesh by converting their world coordinates into UV coordinate using a spherical mapping procedure to get our final refined mesh. As we are not predicting deformation for each of the mesh vertices and only predict a single spatial deformation map, it allows us to sample deformation for many vertices without crippling computational efficiency.

For our GAN training, we tried two variants. For both these variants, we treated the rendered novel views as fake. In one variant, we used input training images as real and in the other rendered images from original view as real. We observed strong improvement over shapes when using input images as real and strong improvement in appearance when using rendered same view images as real.

Evaluation. As there is no ground-truth 3D geometry and texture maps to evaluate, we follow the same evaluation protocol used in DIB-R and evaluate the mask/silhouette projection accuracy. For view generalization, we report FID scores by rendering images from multiple views from the validation set and comparing them against the training data. Finally, we perform user study to evaluate human preferences comparing our method against the baselines.

Results. We provide quantitative and qualitative results in Fig. 6 and Table 1, respectively. From Fig. 6, DIB-R outputs a good prediction for the original view and achieves
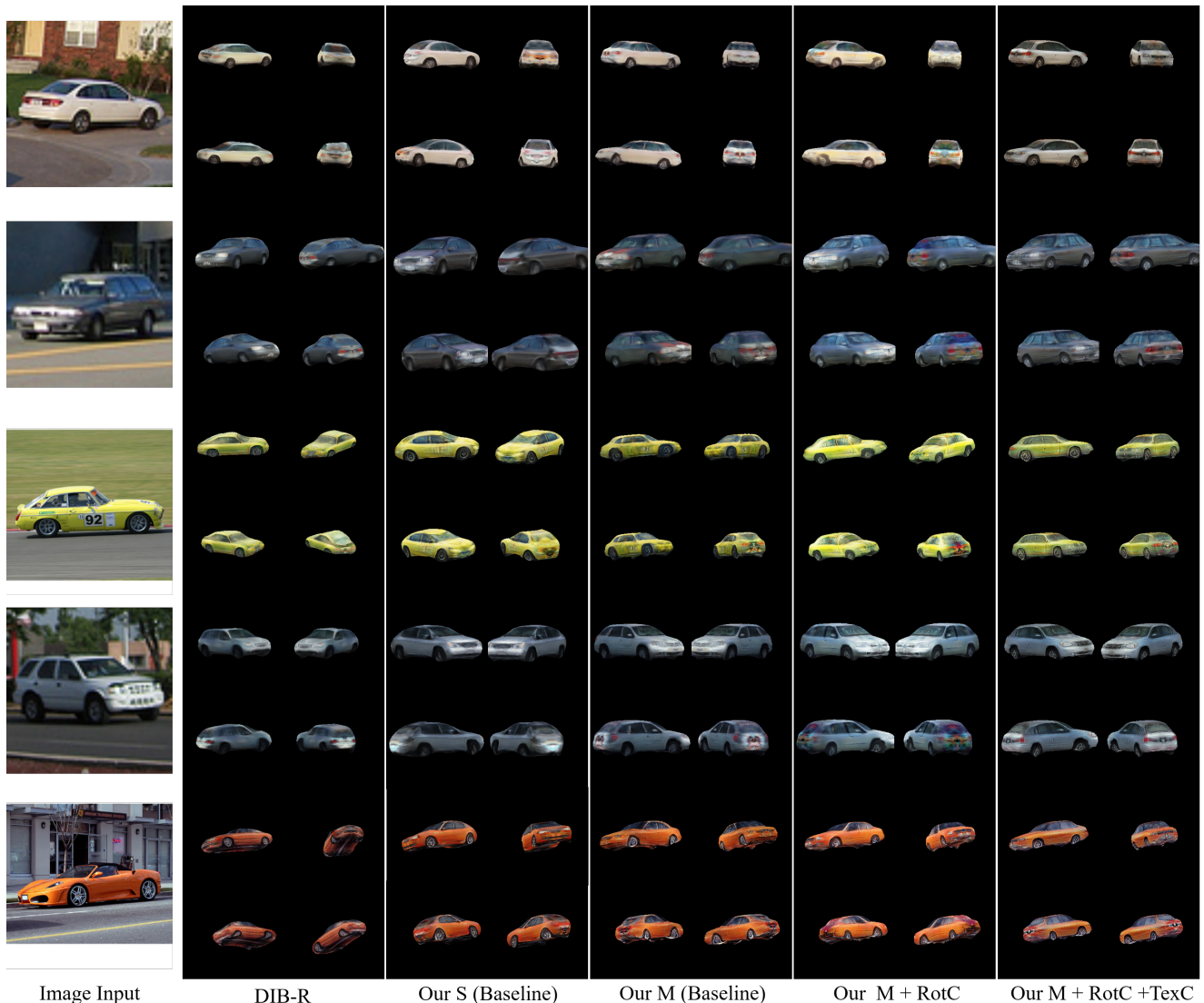
Figure 6: Textured 3D inference on Pascal3D+ cars [42]. We show renderings from 4 different views – the original view and 3 other views by rotating the camera along the azimuth. We provide a comparison with DIB-R [4] and an ablation study showing how each of our components improves overall textured 3D synthesis. DIB-R (second column) suffers from view generalization – the shapes are incoherent and textures are incomplete for novel views. Our single (S) template baseline(third column) using convolutional deformation improves overall shape but is not perfect. Using multiple (M) templates (fourth column), significantly improves the overall geometry of our synthesis. Rotation GAN consistency (RotC; fifth column) losses improve our shape further and also slightly improve texture. In the last column, adding our texture alignment consistency (TexC) loss, we get a strong improvement in the overall texture synthesis. Note that DIB-R uses ground truth (GT) camera for their renderings. The orange car in the bottom row has an erroneous GT camera and suffers from their view during final rendering. Our method reliably produces a correct estimate of the camera, fixing the erroneous GT camera estimations.

a high 2D mIoU (see Table 1). However, their texture and meshes do not generalize across views and results in poor FID scores. Also, their camera is not learned and use ground truth camera for their rendering. Our S (single template) baseline is like DIB-R except the convolutional architecture for the mesh prediction. Our M (multiple templates) baseline improves the geometry considerably, however one can see unrealistic texture completions for novel views. Our

M + RotC (Rotation GAN Cycle Consistency) improves the shape, makes the texture sharper and partially improves the texture predictions for novel views. Our final model with texture alignment consistency (TexC) improves our texture synthesis significantly. In the first row of Fig 6, our final model outputs a realistic front view while others cannot. The realism of our synthesis is also evident from a strong improvement in FID scores (last two rows in Table 1.

For evaluating view generalization, we also conducted a user study on the test set for cars on 220 images. We provide users with input images and predictions rotated in 360°, along the azimuth, in GIF format so they can see the inconsistencies in the shape and texture easily. We ask users to select 3D models that have i) a more realistic texture, ii) a more realistic shape, and iii) overall a better 3D texture synthesis. Both our proposed losses and the convolutional framework significantly outperforms previous methods (see Table 2). Users preferred our method as opposed to DIB-R 59.1% of the time (50% is tie), and as opposed to our baseline without cycle consistency losses 53.54% of the time. The users also preferred our final model when we ask about texture and geometry specifically. We also observe improvement on birds as shown in Fig. 7 and Table 3.

Table 1: Quantitative results on PASCAL3D+ Car dataset [42]. 2D mIOU accuracy predictions and GT, and FID scores between rendered images from various camera viewpoints and training images are reported. We also provide a full ablation study, showcasing improvements over each module. DIB-R [4] uses ground truth camera for their rendering during evaluation for mIOU (shown with *).

Method	Temp.	Deform	RotC	TexC	mIoU	FID
DIB-R [4]	S	Linear			80.0*	198.0
CMR [16]	S	Linear			64.0	–
Ours	S	Conv			77.1	193.7
Ours	M	Conv			76.9	175.5
Ours	M	Conv	✓		**78.0**	171.6
Ours	M	Conv		✓	77.3	166.8
Ours	M	Conv	✓	✓	77.0	**157.6**

Table 2: Human preference evaluation on cars. We report user preferences to our synthesis using proposed convolutional deformation framework (Our Baseline) and additional proposed losses (Our Final) for three different studies. We ask users to choose the synthesis that had better texture, geometry and overall textured 3D synthesis.

	Texture	Geometry	Overall
DIBR vs Our Final	54.1%	66.2%	59.1%
Our Baseline vs Our Final	57.3%	59.6%	53.5%

Table 3: Quantitative results on CUB dataset [41]. 2D mIoU accuracy predictions and GT, and FID scores between rendered images from various camera viewpoints and training images are reported. For DIB-R [4] and CMR [16], we use mIOU reported in [4] which uses a ground truth camera for their evaluation (shown with *) while we use a predicted camera for our mIOU evaluation.

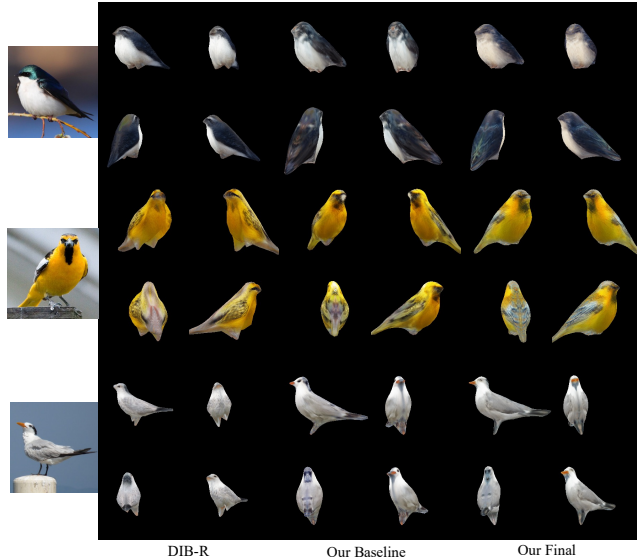| Method	Temp.	Deform	RotC	TexC	mIoU	FID
DIB-R [4]	S	Linear			75.7*	76.0
CMR [16]	S	Linear			73.8*	–
Ours	S	Conv			62.5	76.8
Ours	S	Conv	✓		62.3	75.8
Ours	S	Conv		✓	**62.9**	75.0
Ours	S	Conv	✓	✓	61.7	**70.3**

Figure 7: Textured 3D inference on the CUB dataset [41]. DIB-R renderings are consistent when viewed from original camera, but across novel views there are visible inconsistencies. Our baseline has improvements over shapes, but textures are sometimes not complete. Our proposed losses improve the shape further and produce realistic texture for occluded regions. For example, see feathers at the back for all rendered birds. Our method adds consistent texture for regions even if they are not visible from the original view.

5. Conclusion

In this paper, we introduced a new 3D inference pipeline, principally designed to display good view generalization. We use convolution deformation and show it has controllable and flexible deformation properties and can easily adapt to a particular choice of source image. We also introduce two novel cycle consistency losses responsible for our high-fidelity textured 3D model synthesis from a single 2D image. Our losses enable the sharing of texture information and shape appearances for occluded regions. We show several improvements both qualitatively and quantitatively compared to the recent SOTA method. We hope our work will inspire future works to test their textured 3D inferences for view generalization and depart from 2D mIOU evaluations. Moreover, our overall framework can also apply to recently proposed unsupervised 3D inferences [9, 22] and we believe these approaches would largely benefit from our design choices. Furthermore, because of consistent rendering across views, we believe our method would also be useful for rendering objects [2] into scenes from novel views.

Acknowledgement

References

[1] Aayush Bansal, Shugao Ma, Deva Ramanan, and Yaser Sheikh. Recycle-gan: Unsupervised video retargeting. In *Proceedings of the European conference on computer vision (ECCV)*, pages 119–135, 2018. 3

[2] Anand Bhattad and David A Forsyth. Cut-and-paste neural rendering. *arXiv preprint arXiv:2010.05907*, 2020. 8

[3] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 2, 3, 5

[4] Wenzheng Chen, Huan Ling, Jun Gao, Edward Smith, Jaakko Lehtinen, Alec Jacobson, and Sanja Fidler. Learning to predict 3d objects with an interpolation-based differentiable renderer. In *Advances in Neural Information Processing Systems*, pages 9609–9619, 2019. 1, 3, 4, 5, 6, 7, 8

[5] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*, pages 2172–2180, 2016. 3

[6] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *European conference on computer vision*, pages 628–644. Springer, 2016. 2

[7] Aysegul Dundar, Karan Sapra, Guilin Liu, Andrew Tao, and Bryan Catanzaro. Panoptic-based image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8070–8079, 2020. 3

[8] Aysegul Dundar, Kevin Shih, Animesh Garg, Robert Pottorff, Andrew Tao, and Bryan Catanzaro. Unsupervised disentanglement of pose, appearance and background from images and videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 3

[9] Shubham Goel, Angjoo Kanazawa, and Jitendra Malik. Shape and viewpoint without keypoints. *arXiv preprint arXiv:2007.10982*, 2020. 3, 8

[10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 3

[11] Paul Henderson, Vagia Tsiminaki, and Christoph H Lampert. Leveraging 2d data to learn textured 3d mesh generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7498–7507, 2020. 3

[12] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in neural information processing systems*, pages 6626–6637, 2017. 2

[13] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2016. 3

[14] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017. 3

[15] Subin Jeon, Seonghyeon Nam, Seoung Wug Oh, and Seon Joo Kim. Cross-identity motion transfer for arbitrary objects through pose-attentive video reassembling. *arXiv preprint arXiv:2007.08786*, 2020. 3

[16] Angjoo Kanazawa, Shubham Tulsiani, Alexei A Efros, and Jitendra Malik. Learning category-specific mesh reconstruction from image collections. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 371–386, 2018. 1, 3, 4, 5, 8

[17] Levent Karacan, Zeynep Akata, Aykut Erdem, and Erkut Erdem. Learning to generate images of outdoor scenes from attributes and semantic layouts. *arXiv preprint arXiv:1612.00215*, 2016. 3

[18] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 3

[19] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020. 3

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

[21] Nilesh Kulkarni, Abhinav Gupta, and Shubham Tulsiani. Canonical surface mapping via geometric cycle consistency. In *International Conference on Computer Vision (ICCV)*, 2019. 4

[22] Xueting Li, Sifei Liu, Kihwan Kim, Shalini De Mello, Varun Jampani, Ming-Hsuan Yang, and Jan Kautz. Self-supervised single-view 3d reconstruction via semantic consistency. *arXiv preprint arXiv:2003.06473*, 2020. 3, 8

[23] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In *Adv. Neural Inform. Process. Syst.*, 2017. 3

[24] Shichen Liu, Tianye Li, Weikai Chen, and Hao Li. Soft rasterizer: A differentiable renderer for image-based 3d reasoning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7708–7717, 2019. 3, 4, 5

[25] Yu-Lun Liu, Yi-Tung Liao, Yen-Yu Lin, and Yung-Yu Chuang. Deep video frame interpolation using cyclic frame generation. In *AAAI*, 2019. 3

[26] Matthew M Loper and Michael J Black. Opendr: An approximate differentiable renderer. In *European Conference on Computer Vision*, pages 154–169. Springer, 2014. 3

[27] Dominik Lorenz, Leonard Bereska, Timo Milbich, and Björn Ommer. Unsupervised part-based disentangling of object shape and appearance. In *CVPR*, 2019. 3

[28] Morteza Mardani, Guilin Liu, Aysegul Dundar, Shiqiu Liu, Andrew Tao, and Bryan Catanzaro. Neural ffts for universal texture image synthesis. *Advances in Neural Information Processing Systems*, 33, 2020. 3

[29] Simon Meister, Junhwa Hur, and Stefan Roth. Unflow: Unsupervised learning of optical flow with a bidirectional census loss. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 3

[30] Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong-Liang Yang. Hologan: Unsupervised learning of 3d representations from natural images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7588–7597, 2019. 3

[31] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2337–2346, 2019. 3

[32] Dario Pavllo, Graham Spinks, Thomas Hofmann, Marie-Francine Moens, and Aurelien Lucchi. Convolutional generation of textured 3d meshes. *arXiv preprint arXiv:2006.07660*, 2020. 2, 3

[33] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv preprint arXiv:2007.08501*, 2020. 3

[34] Fitsum A Reda, Deqing Sun, Aysegul Dundar, Mohammad Shoeybi, Guilin Liu, Kevin J Shih, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Unsupervised video interpolation using cycle consistency. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 892–900, 2019. 3

[35] Edgar Schonfeld, Bernt Schiele, and Anna Khoreva. A u-net based discriminator for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8207–8216, 2020. 5

[36] Shubham Tulsiani, Alexei A Efros, and Jitendra Malik. Multi-view consistency as supervisory signal for learning shape and pose prediction. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2897–2905, 2018. 3

[37] Shubham Tulsiani, Tinghui Zhou, Alexei A Efros, and Jitendra Malik. Multi-view supervision for single-view reconstruction via differentiable ray consistency. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2626–2634, 2017. 3

[38] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 52–67, 2018. 5

[39] Ting-Chun Wang, Ming-Yu Liu, Andrew Tao, Guilin Liu, Bryan Catanzaro, and Jan Kautz. Few-shot video-to-video synthesis. In *Advances in Neural Information Processing Systems*, pages 5014–5025, 2019. 3

[40] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8798–8807, 2018. 3

[41] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010. 4, 6, 8

[42] Yu Xiang, Roozbeh Mottaghi, and Silvio Savarese. Beyond pascal: A benchmark for 3d object detection in the wild. In *IEEE winter conference on applications of computer vision*, pages 75–82. IEEE, 2014. 4, 6, 7, 8

[43] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *International Conference on Machine Learning*, pages 7354–7363. PMLR, 2019. 3

[44] Bo Zhao, Lili Meng, Weidong Yin, and Leonid Sigal. Image generation from layout. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019. 3

[45] Zhedong Zheng, Xiaodong Yang, Zhiding Yu, Liang Zheng, Yi Yang, and Jan Kautz. Joint discriminative and generative learning for person re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2138–2147, 2019. 3

[46] Hang Zhou, Jihao Liu, Ziwei Liu, Yu Liu, and Xiaogang Wang. Rotate-and-render: Unsupervised photorealistic face rotation from single-view images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5911–5920, 2020. 3

[47] Keyang Zhou, Bharat Lal Bhatnagar, and Gerard Pons-Moll. Unsupervised shape and pose disentanglement for 3d meshes. *arXiv preprint arXiv:2007.11341*, 2020. 3

[48] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017. 3

[49] Jun-Yan Zhu, Zhoutong Zhang, Chengkai Zhang, Jiajun Wu, Antonio Torralba, Josh Tenenbaum, and Bill Freeman. Visual object networks: Image generation with disentangled 3d representations. *Advances in Neural Information Processing Systems*, 2018. 3

[50] Xiangyu Zhu, Xiaoming Liu, Zhen Lei, and Stan Z Li. Face alignment in full pose range: A 3d total solution. *IEEE transactions on pattern analysis and machine intelligence*, 41(1):78–92, 2017. 3