Improving Image-based Generation of Implicit Texture Fields for 3D Objects

Vikas Thamizharasan Joshua Pierce Daniel Ritchie

Brown University

Abstract

Transferring the surface texture from an image of an object to a 3D model of an object is a long-standing problem at the intersection of vision and graphics. One new technology that has recently been applied to this problem is neural implicit fields: learning a neural network which takes as input the 3D object to be textured, the image, and a point in space, and returns an RGB texture value at that point. However, current texture field methods produce low-frequency output textures which do not faithfully represent the input image. In this paper, we present an improved method for generating implicit texture fields based on an input image. We leverage the power of modern single-image 3D reconstruction methods to unproject pixels from the conditioning image into points in 3D. As the image object may be geometrically different from the 3D object, we leverage unsupervised implicit dense correspondences to warp these points toward the shape of the 3D object. Our method then makes use of these points in two ways. First, it defines an initial texture field as an average of nearby image points, which gives high quality texture in regions that are visible in the input image. We improve coverage further by evaluating distances in a feature space produced by a shape segmentation network, in which points on semantically-related parts of an object (e.g. symmetric parts) tend to be close. To produce texture in uncovered regions, it then trains a neural implicit network to predict a residual texture field on top of the initial field. We compare our method to the prior state-of-the-art in image-based implicit texture field generation, showing that it produces texture fields that more faithfully reflect the input image and adhere to the object's local geometry.

1. Introduction

Demand for high-quality 3D models is increasing in many fields: from games and virtual reality, to interior design and retail, to synthetic training data for computer vision and robotics applications. Creating this content from scratch is time-consuming and requires considerable expertise. Thus, graphics and vision researchers are seeking new data-driven techniques to augment human abilities and accelerate the process.

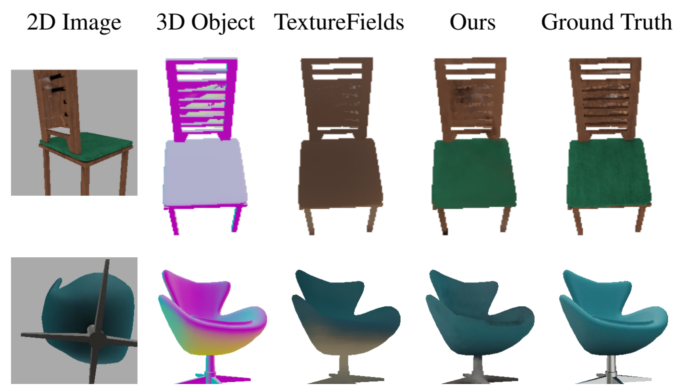

Figure 1. Given an input 3D object and a single image depicting an object of the same semantic class, our method transfers the appearance of the object in the image to the input 3D object by generating an implicit texture field.

One problem of interest is how to ease the process of applying surface texture to 3D objects. An appealing idea is to develop a method that can automatically transfer texture from a single reference image of an object to a 3D model of the same type of object. This is an ill-posed problem: the system has to understand the 3D structure of the object depicted in the image, deal with shape differences between the object in the image and the object to be textured, rectify distortions in textures and 'hallucinate' plausible texture in regions occluded in the input image.

Recently, the surge in popularity of neural implicit fields as representations for computer vision problems has led to their application in image-based texture transfer. Given an input reference image, an input shape to be textured, and a point in space in the coordinate frame of that object, a neural network can predict a color at that point [9]. This implicit field formulation is appealing, as it avoids the need to define a consistent surface parameterization of the object (which is itself still something of an open research problem, in general). However, prior work on implicit texture fields has been limited to producing blurry, low-frequency surface texture with poor part-level correspondences between the image and the input mesh.

In this paper, we present an improved system for generating an implicit texture field from a single image. Our key insight is that rather than task a neural network with predicting an entire texture field from an image, it is better to use knowledge of the image formation process to take texture directly from the image where possible, and only task a network with predicting what remains to be filled in. To that end, we leverage recent advances in single-image 3D reconstruction (specifically, Normalized Object Coordinate Space (NOCS) maps [16]) to unproject 2D image pixels to 3D colored points. To account for shape differences between the input image and object to be textured, our system computes a dense correspondence between the unprojected image points and the object, and warps the former toward the latter. Armed with an aligned set of colored image points and object points, our system produces an implicit texture field via a two-step process. First, it defines an initial non-parametric texture field whose value at any point is an average of nearby image point colors. This field provides accurate texture for regions of the object covered by visible image regions. We further improve this coverage by computing distances not in $\mathbb{R}^3$-space but in a higher-dimensional feature space where points on semantically-related parts of the object are closer though they may be further away in Euclidean space (e.g. points on two different arms of a chair). Second, our system fills in the remaining uncovered regions of the object using a learned neural texture field network. This module is designed as a residual network, predicting the difference in color that should be added to the output of the first non-parametric field.

We evaluate our system by comparing it to the prior state-of-the-art, TextureFields, on generating texture for 3D chair objects from the 3D FUTURE dataset [3]. Our system quantitatively outperforms TextureFields on the task of recovering the ground-truth textures of these objects, and it also produces qualitatively better textures for novel image-to-object texture transfer settings for which there is no ground truth.

In summary, our contributions are:

- A system for a non-parametric texture field constructed directly from input pixels with a neural implicit field via a residual prediction approach.

2. Related Work

Some prior work has addressed the problem of generating textures on 3D objects based on an image. One work in this area detects rectified patches in the input image and extracts illumination-corrected textures from them to be transferred to the object [17]. This is limited to conditioning images with homogeneous texture patterns, uv-parameterized input mesh, assumes input mesh and geometry of conditioning image are similar and does not have any data-driven prior for occluded regions. The TM-NET [4] system is designed to generate novel textures without image conditioning, but can be adapted for synthesis given a single-view conditioning image. However, it has only demonstrated relatively homogeneous textures (e.g. wood grain). Another system takes renderings of a synthetic object as input, transforms each rendering to photorealistic appearance using cycle-consistent GANs, and then uses a differentiable renderer to consolidate these images into a texture map for the object [10]. The authors demonstrate the ability to condition the model on the dominant color that the output should have, but it cannot be conditioned on the complete appearance of an object from another image. This method also requires an explicit surface parameterization of the object to be textured. A somewhat related work proposes a generative model of textured meshes which can be learned only from 2D image data [5]. In principle, one could transfer the texture maps it produces for its synthesized meshes to another target object; however, this system is limited to genus zero shapes and also requires an explicit surface parameterization. As training 2D CNNs on spatially inconsistent uv maps can lead to irregular receptive fields, works like TextureNet [6] have proposed neural architectures to operate on locally consistent patches. But these have not been applied to problems in texture transfer or synthesis.

Our work is most closely related to that of TextureFields, which proposed the first neural implicit field for producing 3D object textures [9] (either conditioned on a single-view image or unconditioned). The method displays an impressive ability to transfer surface appearance features from a 2D image of one object to the 3D geometry of another, including producing texture in occluded regions that is plausible and consistent with the visible regions. However, its output textures are low-frequency, sometimes reducing to little more than a single average color for the entire object.

Other recent work has explored neural implicit fields for modeling appearance in addition to geometry. The PIFu sytsem aims at reconstructing the same geometry and appearance from the input image, whereas we seek to transfer input image appearance to a different 3D geometry [11]. pi-GAN [1], scene representation networks [12], and neural radiance fields [8] can also represent textured objects via neural implicit fields. They all produce representations which must be rendered via some form of volumetric ray marching, whereas we seek a surface texture field.

3. Approach

Figure 2 shows an overview of our system. As input, it takes a 3D object to be textured (represented as a point cloud) and a single-view *conditioning image* depicting an object of the same class from which texture should be borrowed. As output, the system produces an *implicit texture field*, ie. a continuous function $f : \mathbb{R}^3 \to \mathbb{R}^3$ which maps a point in space to an RGB color.

We divide our system into three stages. In the first stage, the pixels from the conditioning image are unprojected to points in the same 3D canonical space as the input object, and then they are warped to align with the input object's points. The output of the first stage is a merged point cloud consisting of both input object points and colored conditioning image points.

The second stage then produces an initial *non-parametric texture field* via averaging the values of nearby conditioning image points for any given query point. This non-parametric field is not well-defined in regions of space that are occluded in the input conditioning image. To improve the coverage of this field, we search for nearest neighbors in a high-dimensional feature space produced by a pre-trained shape segmentation network, in which points on semantically-related shape parts are close (e.g. two arms of an armchair). Evaluating distances in this space allows for texture to be meaningfully interpolated to some regions which are not visible in the conditioning image.

Finally, the third stage deals with regions of the input object that remain uncovered by the non-parametric texture field from stage two. No content from the conditioning image could be applied to these regions, so we instead rely on a data-driven prior learned from a large dataset of textured 3D objects to 'hallucinate' plausible texture here. Specifically, we train a neural implicit field model to output texture for points in space, given the conditioning image and object to be textured as input. We train the network to output a *residual field* of differences from the non-parametric texture field of stage two, allowing the network to correct issues with that field as well as fill in occluded regions.

The following sections describe each of these three stages in more detail.

4. Lifting Input Image Pixels to 3D

Prior work on image-based implicit texture field generation applied 2D image-space processing to the conditioning image to produce texture features which are then combined with geometric features from the input object to be textured [9]. We hypothesize that image-based implicit textured field generation is better solved by first lifting the 2D conditioning image into 3D and jointly processing this information with the input object geometry in the same space.

To perform this 2D to 3D lifting, we use a normalized object coordinates (NOCS) map of the conditioning image [16]. A NOCS map for an image contains the 3D coordinates of each pixel in a category-canonical coordinate space. In our experiments, we assume that the input object to be textured is provided in this coordinate system, though it is possible to 'canonicalize' objects that are not provided as such [16]. We also use ground-truth NOCS maps for our conditioning images, though again, it is possible to learn to predict them with high accuracy [16] [13]. The NOCS map also implicitly provides a foreground/background segmentation, which we use to lift only foreground pixels to 3D points.

Because the conditioning image may depict an object of different geometry from the input object to be textured, we cannot directly merge these unprojected pixels with points from the input object, as they may not align. Instead, we must first warp the conditioning image points to align with the input object points. We accomplish this warp by computing a dense correspondence between the conditioning image points and the input object points using ShapeFlow [7]. Specifically, we perform latent space optimization to embed both these point clouds into ShapeFlow's learned latent space and deform both source (conditioning image points) and target points (input object points) to the zero latent code corresponding to the hub, which is an aligned canonical space giving us dense correspondence. Figure 3 shows an example of this correspondence-based warping.

5. Non-parametric Field for Visible Regions

The output of the previous stage is a point cloud consisting of points from both the input object and the conditioning image. The approximately-aligned conditioning image points provide a strong signal for what texture our output implicit texture field should contain, at least in the the regions of space corresponding to visible regions in the image. Thus, this stage constructs an initial implicit texture field by interpolating between the colors of the conditioning image points. As this field requires no learnable parameters, we call it the *non-parametric field*. We define the non-parametric field by averaging nearby image points. Specifically, for a query point $\mathbf{q}$, the texture field is defined as the unweighted average of the colors of the five nearest neighbor points in the conditioning image point cloud (see Figure 5).

This field has well-defined texture only in regions of space which are close to at least one of the image points. We can increase the coverage of this field by changing our distance metric for nearest neighbor lookup to one in which spatially-distant points can still be close if they are semantically related. For example, two symmetric parts of an object very likely should have the same texture: even if they are

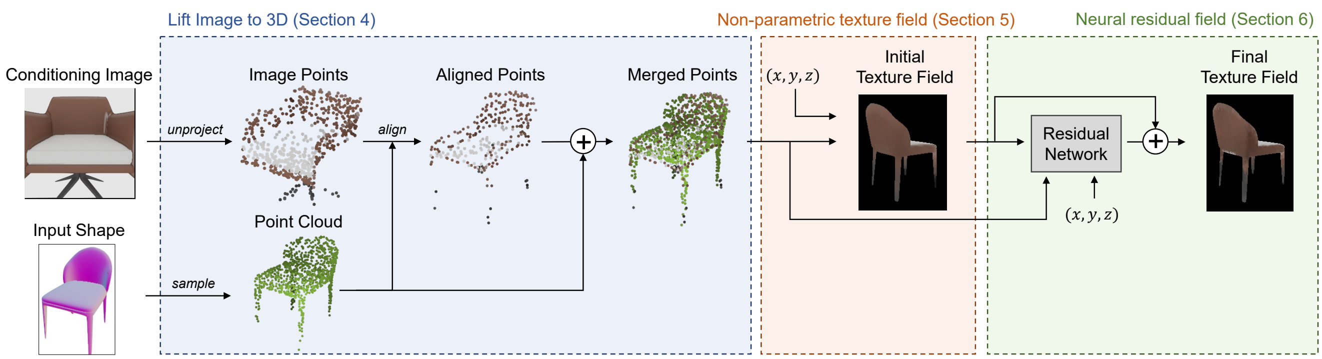

Figure 2. An overview of our approach. Given an input 3D object and conditioning image, we first lift the input image pixels to colored 3D points and warp them to align with the geometry of the input object. These points are then used to define an initial, non-parametric texture field based on averaging their values. Since this initial field only defines texture for regions of space visible in the conditioning image, we also train a neural residual texture field to fill in these regions.

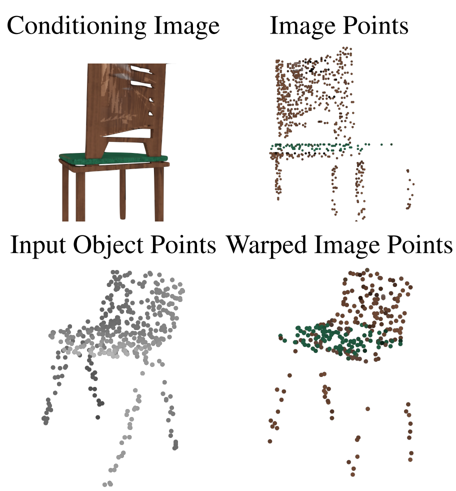

Figure 3. The effect of warping conditioning image points using dense correspondences from ShapeFlow. Clockwise from top left: the conditioning image, the unprojected 3D points from the image, unprojected points after being warped toward the input object, the input object.

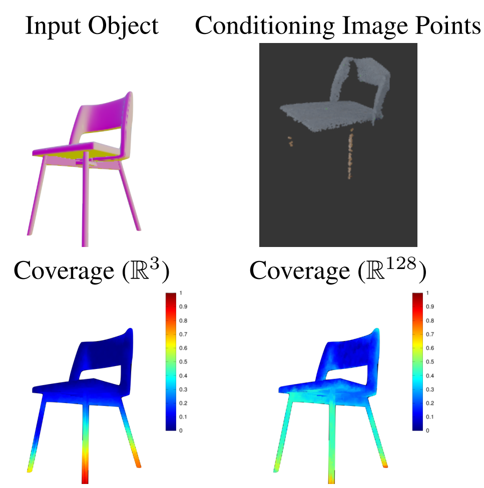

Figure 4. Visualizing how 'covered' the input object is by the conditioning image points under different distance metrics ($\mathbb{R}^3$ Euclidean space or the $\mathbb{R}^{128}$ space of a pre-trained shape classifier). In the coverage images, points on the object are colored by their distance to their nearest conditioning image point.

spatially far apart, we would like the non-parametric field to interpolate texture from points on one part to its symmetric counterpart (especially if the counterpart is not visible in the conditioning image and has no image points near it).

To construct such a distance metric, we lift points to a higher dimensional feature space in which Euclidean distance has the properties we desire. To do this, we pass all points through a DGCNN [18] pre-trained on a semantic part segmentation task and take the penultimate (third layer) features as $\mathbb{R}^{128}$ per-point descriptors. Figure 4 compares how much of the input object is 'covered' by nearby conditioning image points in 3-space vs. the 128-dimensional DGCNN feature space. Distances are not directly comparable between the two spaces: points become sparser in higher-dimensional spaces, and thus average point-to-point distances are higher. Still, it is clear that the regions of the chair which are severely uncovered (the feet) are reduced in the higher dimensional space.

6. Neural Residual Field

As shown in Figure 4, some regions of the input object may remain uncovered by any conditioning image point, even when distances are evaluated in the DGCNN feature space. For instance, this happens in cases where there is an occluded part of the input object that is not semantically related to any visible region in the conditioning image (e.g. a

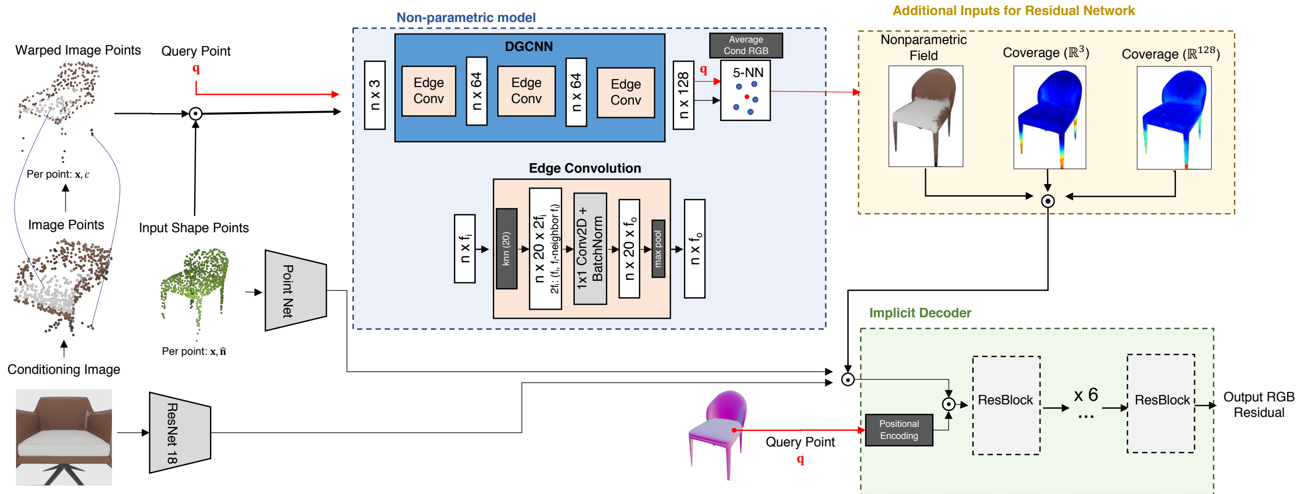

Figure 5. Our pipeline for generating implicit texture fields from images. It takes as input a conditioning image and shape points sampled from an object to be textured. Pixels from the conditioning image are unprojected to 3D space using NOCS maps [16] and then warped to align with the input shape points using ShapeFlow [7]. These points are then passed to our non-parametric texture field model, which lifts them to a higher dimensional space using a pre-trained segmentation DGCNN and then uses nearest neighbor lookups in this space to define the color at any give query point $\mathbf{q}$. The non-parametric model also provides two coverage maps indicating how well the conditioning image overlaps with each point on the shape in 3-space and and higher-dimensional feature space. Finally, the outputs of the non-parametric field are passed to an implicit decoder which estimates the residual between the non-parametric model's predicted color and the ground truth.

top-down image of a chair does not show any of the chair's legs). To texture such regions, we instead rely on priors learned from data. Specifically, we learn a *neural residual field* that predicts differences from the non-parametric field. The residual field uses the same architecture as the implicit field network from TextureFields [9], which takes as input a point in space and returns an RGB color. We concatenate to the input point location $\mathbf{x}$ some additional inputs: (1) the color predicted by the non-parametric field at $\mathbf{x}$, (2) the $\mathbb{R}^3$ coverage at $\mathbf{x}$, and (3) the $\mathbb{R}^{128}$ coverage at $\mathbf{x}$. We also transform $\mathbf{x}$ into a positional encoding using a Fourier feature transform [15]. The network is then trained to predict the difference between the value of the non-parametric field at $\mathbf{x}$ and the ground-truth texture value at that point. Like TextureFields, we train only on points sampled from the object's surface, as ground truth texture values are only defined there.

7. Results & Evaluation

Here we evaluate how well our model can transfer textures from 2D images to 3D shapes, investigate ablations, and compare with the prior state of the art. All experiments were run on a NVIDIA Quadro RTX 6000 and a GeForce RTX 3090 both with 24GB RAM.

Dataset. For our experiments, we use 3D chair models from the the 3D FUTURE Dataset [3], a dataset of high-quality textured meshes of furniture objects. Prior works have used ShapeNet [2] where textures are generally homogeneous and lack detail. We use a total of 968 chairs (700 train, 140 validation, 128 test). We pre-process these meshes into inputs for our system by transforming them into the NOCS space [16] and then uniformly sampling 2048 points from the surface where each sample consists of a point location, surface normal, and surface color. We render conditioning images for these meshes by sampling camera viewpoints around the viewing hemisphere resulting in pairs for ground truth supervision. We additionally render NOCS, normals, and ground truth color maps of the input mesh offline from various view points for visualization at inference time, avoiding expensive online rendering. These maps are also used for quantitative evaluations as discussed in the next section.

Method	L1 ↓	L2 ↓	LPIPS ↓	SSIM ↑
TextureFields	0.0316	0.0059	0.1246	0.9203
TextureFields + PE	0.0302	0.0061	0.1187	0.9216
Non-parametric Only	0.0325	0.0077	0.1139	0.9122
Residual	0.0292	0.0054	0.1093	0.9210
Residual + PE (Ours)	0.0281	0.0052	0.1053	0.9228

Table 1. Quantitative comparison between different model variants on the task of reconstructing the texture of an object given a single rendering of that object. Gold/silver/bronze colors indicate 1st/2nd/3rd-best performing models for each metric.

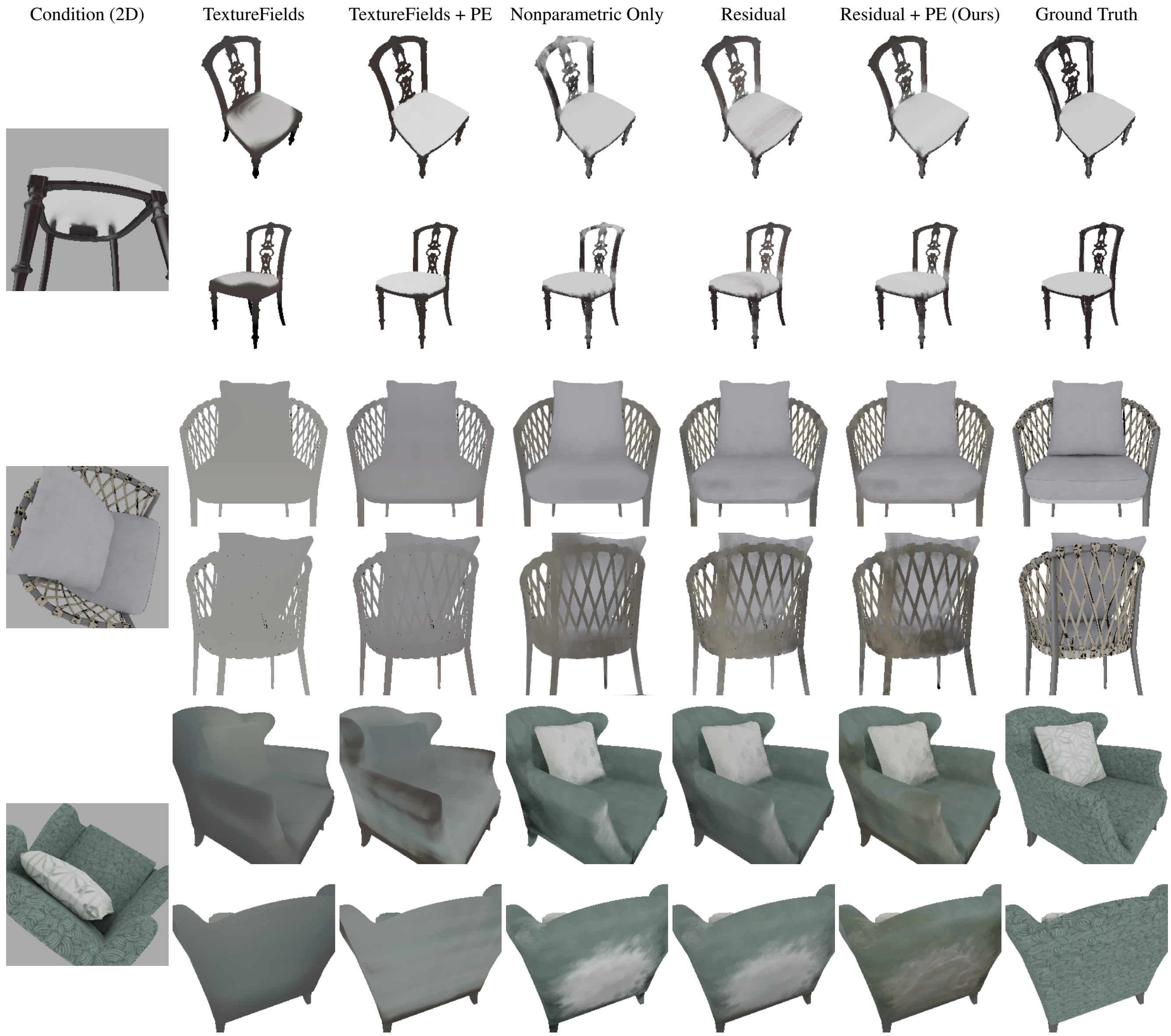

Figure 6. Qualitative comparison between different texture field methods in cases where the conditioning image is a rendering of the input object.

Evaluation protocol. To compare different methods, we use both qualitative comparison of output texture fields as well as quantitative evaluation. To perform quantitative evaluation, we perform experiments where the conditioning image is a view of the input object, so we have a known ground truth texture field for that object. We render the predicted textures from 10 viewpoints randomly sampled on the hemisphere and measure the model's performance by computing four metrics: L1 distance, L2 distance, SSIM, and LPIPS [19] between the rendered predicted image and rendered ground truth image.

Experiments. Using these metrics, we compare the following conditions:

- *TextureFields:* The implicit field model from TextureFields [9].
- *TextureFields + PE:* Adding input positional encodings to TextureFields.
- *Nonparametric Only:* Producing output texture using only our non-parametric implicit field.
- *Residual:* The non-parametric field plus the residuals predicted by our residual field network.
- *Residual + PE:* The residual field model with input

Figure 7. Qualitative comparison between different texture field methods in cases where the conditioning image depicts a different object than the input object. In the third example, we also show a zoomed in section demonstrating the importance of dense correspondence - which prevents color bleeding into different parts.

positional encodings.

Table 1 shows quantitative results. Our method dominates TextureFields on all metrics. While positional encodings are an important component, the residual model without positional encoding also outperforms the TextureFields. Figure 6 shows qualitative results from this experiment. While TextureFields exhibits some learned priors (e.g. by correctly coloring the occluded chair-back in the first row), it fails to reflect high-frequency details seen in the conditioning image. By comparison, the textures estimated by both residual models more accurately reflect these details while still leveraging learned priors. More qualitative results are provided in supplemental.

We also conduct qualitative evaluation on examples where the conditioning image depicts a different object than the input object. In this case, we cannot conduct quantitative comparisons because there is no ground truth. For this experiment, we consider another condition, Residual + PE (No Def), which omits the step of aligning the conditioning image points to the input object using ShapeFlow deformations. Figure 7 shows this qualitative comparison. In the first example, both the non-parametric and residual models transfer the distinct colors of the conditioning chair's back and seat, highlighting the strength of the non-parametric model's use of the part-aware feature-space. In the second example, the residual field is critical for correcting the ap-

Figure 8. Typical failure cases of our method.

pearance of the chair back. Both the second and third examples illustrate the importance of aligning the conditioning image points with the input shape points using ShapeFlow deformations, preventing color leaking from the chair seat to the back (2nd example) or from the seat to the legs (3rd example). More qualitative results are provided in supplemental.

8. Conclusion

In this paper, we presented a new system for image-based implicit texture field generation that improves upon the prior state of the art. Our system leverages single-image 3D reconstruction to lift conditioning image pixels to the same 3D space as the object to be textured. It makes use of these 3D colored points in two ways: first defining a non-parametric texture field through nearest-neighbor averaging, and then completing occluded regions using a neural residual field. On the challenging task of texturing chairs from the 3D FUTURE dataset, our system outperforms prior work.

Our system does still have some limitations. Figure 8 shows some typical failure cases. While our model captures more high-frequency detail than TextureFields, it still exhibits smoothing for very high frequencies (top row). In addition, regions of the input object that are both occluded in the input image and infrequent in the training data can be hard to synthesize plausible texture for, as neither the image-based non-parametric field nor the data-driven residual field have much information to draw on (e.g. the small chair pillow in the bottom row).

In our current neural residual model, the implicit decoder takes global geometric and conditioning-image features as input. One interesting avenue of further research could entail generating local features from the conditioning-image and shape data to condition the implicit texture field. One could image computing these features jointly, by processing both input shape points and unprojected image points in a common representation space; an approach along these lines has been successful in learning to segment large-scale point clouds [14]. Another avenue of further study would be to incorporate a more comprehensive loss-function than the current pixel-wise approach allows. For example, incorporating an adversarial loss may address some of the asymmetric, unrealistic artifacts in the generated textures.

References

[1] Eric Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In *arXiv*, 2020. 2

[2] Angel X. Chang, Thomas A. Funkhouser, Leonidas J. Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, L. Yi, and Fisher Yu. Shapenet: An information-rich 3d model repository. *ArXiv*, abs/1512.03012, 2015. 5

[3] Huan Fu, Rongfei Jia, Lin Gao, Mingming Gong, Binqiang Zhao, Stephen J. Maybank, and Dacheng Tao. 3d-future: 3d furniture shape with texture. *Int. J. Comput. Vis.*, 129:3313–3337, 2021. 2, 5

[4] Lin Gao, Tong Wu, Yu-Jie Yuan, Ming-Xian Lin, Yu-Kun Lai, and Hao Zhang. Tm-net: Deep generative networks for textured meshes. *ACM Transactions on Graphics (TOG)*, 40(6):263:1–263:15, 2021. 2

[5] Paul Henderson, Vagia Tsiminaki, and Christoph Lampert. Leveraging 2D data to learn textured 3D mesh generation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[6] Jingwei Huang, Haotian Zhang, Li Yi, Thomas Funkhouser, Matthias Nießner, and Leonidas J Guibas. Texturenet: Consistent local parametrizations for learning from high-resolution signals on meshes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4440–4449, 2019. 2

[7] Chiyu Jiang, Jingwei Huang, Andrea Tagliasacchi, and Leonidas Guibas. Shapeflow: Learnable deformations among 3d shapes. In *Advances in Neural Information Processing Systems*, 2020. 3, 5

[8] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 2

[9] Michael Oechsle, Lars Mescheder, Michael Niemeyer, Thilo Strauss, and Andreas Geiger. Texture fields: Learning texture representations in function space. In *Proceedings IEEE International Conf. on Computer Vision (ICCV)*, 2019. 2, 3, 5, 6

[10] Amit Raj, Cusuh Ham, Connelly Barnes, Vladimir Kim, Jingwan Lu, and James Hays. Learning to generate textures on 3d meshes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019. 2

[11] Shunsuke Saito, , Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. *arXiv preprint arXiv:1905.05172*, 2019. 2

[12] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. In *Advances in Neural Information Processing Systems*, 2019. 2

[13] Srinath Sridhar, Davis Rempe, Julien Valentin, Sofien Bouaziz, and Leonidas J. Guibas. Multiview aggregation for learning category-specific shape reconstruction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 3

[14] Hang Su, Varun Jampani, Deqing Sun, Subhransu Maji, Evangelos Kalogerakis, Ming-Hsuan Yang, and Jan Kautz. SPLATNet: Sparse lattice networks for point cloud processing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2530–2539, 2018. 8

[15] Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *NeurIPS*, 2020. 5

[16] He Wang, Srinath Sridhar, Jingwei Huang, Julien Valentin, Shuran Song, and Leonidas J. Guibas. Normalized object coordinate space for category-level 6d object pose and size estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2, 3, 5

[17] Tuanfeng Y. Wang, Hao Su, Qixing Huang, Jingwei Huang, Leonidas Guibas, and Niloy J. Mitra. Unsupervised texture transfer from images to model collections. *ACM Trans. Graph.*, 35(6), Nov. 2016. 2

[18] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic graph cnn for learning on point clouds. *ACM Trans. Graph.*, 38(5), oct 2019. 4

[19] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 6