# Implicit Feature Networks for Texture Completion from Partial 3D Data

Julian Chibane and Gerard Pons-Moll

Max Planck Institute for Informatics
{jchibane,gpons}@mpi-inf.mpg.de

**Abstract.** Prior work to infer 3D texture use either texture atlases, which require uv-mappings and hence have discontinuities, or colored voxels, which are memory inefficient and limited in resolution. Recent work, predicts RGB color at every XYZ coordinate forming a texture field, but focus on completing texture given a single 2D image. Instead, we focus on 3D *texture* and geometry completion from partial and incomplete *3D scans*. IF-Nets [2] have recently achieved state-of-the-art results on 3D geometry completion using a multi-scale deep feature encoding, but the outputs lack texture. In this work, we generalize IF-Nets to texture completion from partial textured scans of humans and arbitrary objects. Our key insight is that 3D texture completion benefits from incorporating local and global deep features extracted from *both* the 3D partial texture and completed geometry. Specifically, given the partial 3D texture and the 3D geometry completed with IF-Nets, our model successfully in-paints the missing texture parts in consistence with the completed geometry. Our model won the SHARP ECCV'20 challenge, achieving highest performance on all challenges.

**Keywords:** implicit representation, implicit function learning, texture field, implicit feature networks, texture completion, 3D reconstruction, human reconstruction

## 1 Introduction

Motivated by difficulties in the 3D capturing process of complete scans the ECCV 2020 SHApe Recovery from Partial textured 3D scans (SHARP) Workshop[1] initialized a challenge on reconstructing full 3D textured meshes from partial 3D scan acquisitions. Complete 3D scanning, amongst other challenges, need to capture varying levels of details from the scene, need to handle occlusions, transparency, movement and require bulky studio setups. On the other hand, mobile hand-held scanners[2] allow an adaptive usage. Longer acquisition times, can result in partial scans with inaccuracies which we target to complete in this work.

[1] https://cvi2.uni.lu/sharp2020/
[2] Artec3D, https://www.artec3d.com

For the geometry reconstruction we rely on the state-of-the-art Implicit Feature Networks [2] (IF-Nets) neural 3D processing architecture. We find exciting results for reconstruction of humans: even in the absence of whole body parts, as arms or feet, in the corrupted input, the fully learned IF-Nets are generating plausible human shape completions, we expected to require a human body model.

However, the reconstructions are missing texture - an integral part of a human scan, consumers are interested in. There are two common texture representation which both have limitations: 1) texture can be represented as 2D texture atlases stored as images, which need geometry templates to find uv-mappings from 3D to 2D and have hard to handle discontinuities and 2) colorized voxel grids, which are memory inefficient and restrict to low-frequency texture. A novel texture representation, learned texture fields, predict rgb color at a specific xyz coordinate. However, prior work do inference only based on 2D image evidence [9, 10, 15, 8]. In this work we extend the IF-Net architecture for use of texture completion from colored, partial 3D data: humans and arbitrary objects. As input to our texture field prediction we use the partial textured input surface and an IF-Net untextured geometry completion, and feed both as voxelized grids into our network. For every point on the input full geometry we predict the rgb color. We find our model successfully inpaints the missing texture parts in consistence with the completed geometry.

## 2 Related Work

*Implicit Function Leaning (IFL)* IFL methods use either binary occupancies [6, 3, 2, 15] or signed distance functions [11, 1, 7, 4] as shape representation for learning. These methods incorporate a simple but very powerful trick: a network predicts occupancy or the SDF valuea at continuous point locations ($x$-$y$-$z$).
IF-Nets[2] showed state-of-the-art performance on reconstruction from partial 3D. However, the reconstructions are missing texture. We therefor, in this work, extend IF-Nets to reconstruction of complete, textured reconstructions.
Another application of IFL methods is novel view synthesis, that is, generating novel view-points given one or multiple images [16, 8], however, this is not the focus of our work.

*Texture fields* Recent work, predicts RGB color at every XYZ coordinate forming a texture field, but focus on completing texture given a single 2D image [15, 10] or multiple 2D images [8, 9]. Instead, we focus on 3D *texture* and geometry completion from partial and incomplete *3D scans*, which is required for the given challenge.

## 3 Method

Our goal is to reconstruct a completed textured surface of humans and objects given only a textured partial observation. To predict the full, textured recon-

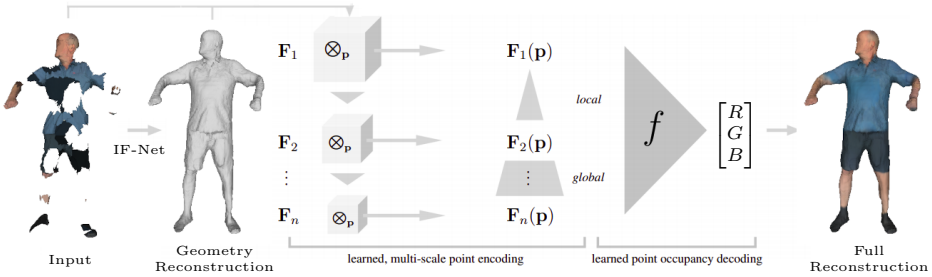


**Fig. 1.** Overview of inference-time prediction of textured, complete surfaces from corrupted, partial input. We train a standard IF-Net to reconstruct untextured geometry given the corrupted, partial input. An rgb value at point $\mathbf{p}$ is predicted by explicitly extracting local and global deep features computed from *both* the 3D partial texture and completed geometry. Our key insight is that this enables our model to generate texture in consistence with the completed geometry.

struction we extend IF-Nets [2] to inference of texture fields. (Sec. 3.1) To obtain geometry reconstructions we train a regular IF-Net (Sec. 3.2).

### 3.1 Texture Inference

We learn to complete the texture of a given partial 3D surface, $\mathcal{T}_i$. For this, we learn a texture field, that is, a mapping of points $\mathbf{p}$, on a complete but untextured surface, to a color at that point. As input $\mathbf{x}$ to the texture field prediction, we additionally use a complete untextured surface geometry of the object. During training the complete surface geometry is provided by the ground truth surfaces, $\mathcal{S}_i$, during inference it is predicted as described in Sec 3.2.

*Encoding* We construct an input encoding *aligned* with the input data. That is, using a 3D CNN we neurally process the input to feature grids, that lie in the same Euclidean space as the input. By recursively applying $n$ times, 3D convolutions followed by down scaling through max pooling, we create *multi-scale deep feature grids* $\mathbf{F}_1, .., \mathbf{F}_n$. The feature grids at the early stages capture texture and shape details, whereas feature grids at the late stages have large receptive fields and capture global structure, for example to segment garments for giving them a coherent texture. See [2] for details.
To adapt the procedure for texture inference, we feed 4 channels of input data into the encoder, 3 channels for the textured partial observation - one channel per rgb - and the fourth channel encodes the complete surface geometry. For usage with a 3D CNN we need to voxelize the surfaces, we do this by densely sampling surface points and marking their nearest neighbor in an enclosing voxel grid. For the partial textured input, we mark the nearest neighbour voxels by the rgb intensities in $\{0, \ldots, 255\}$ and all other with $-1$. For the complete surface geometry, we mark voxels selected as nearest neighbours with 1 and all other

with 0. We denote the encoder as $g(\mathbf{x}) := \mathbf{F}_1, .., \mathbf{F}_n$, where our input is denoted as $\mathbf{x}$.

*Decoding* We create a point $\mathbf{p}$ specific encoding $\mathbf{F}_1(\mathbf{p}), .., \mathbf{F}_n(\mathbf{p})$ by extracting deep features at the location $\mathbf{p}$ from the global encoding $g(\mathbf{x}) = \mathbf{F}_1, .., \mathbf{F}_n$. This is possible because $g(\mathbf{x})$ is kept aligned with Euclidean space. Since our feature grids are discrete, we use trilinear interpolation to query continuous 3D points $\mathbf{p} \in \mathbb{R}^3$. See [2] for details.
The point encoding $\mathbf{F}_1(\mathbf{p}), .., \mathbf{F}_n(\mathbf{p})$, is then fed into a point-wise decoder $f(\cdot)$, parameterized by a fully connected neural network with ReLU activations, to regress the rgb color at $\mathbf{p}$.

*Training* At training time we use the textured partial surfaces, $\mathcal{T}_i$. and ground truth complete surface, $\mathcal{S}_i$, without texture, as inputs for the network. The task of the network is then, to learn to regress the correct color $\text{rgb}(\mathbf{p}, \mathcal{S}_\text{i})$ at a surface point $\mathbf{p} \in \mathcal{S}_i$ of the ground truth surface. We construct training data by sampling points $\mathbf{p}$ uniform at random on the surface $\mathcal{S}_i$ and find their ground truth $\text{rgb}(\mathbf{p}, \mathcal{S}_\text{i})$ values. More specifically, we have $\mathcal{S}_i$ available as mesh .obj-files with attached texture atlases. Therefore, to find $\text{rgb}(\mathbf{p}, \mathcal{S}_\text{i})$, we check for the triangle $\mathbf{p}$ is on, find the uv-coordinates of $\mathbf{p}$ by Barycentric interpolation of the uv-coordinates of the triangle vertices and extract the rgb value from the texture atlas at the found uv location. We use an L1 mini-batch loss:

$$\mathcal{L}_\mathcal{B}(\mathbf{w}) := \sum_{\mathbf{x} \in \mathcal{B}} \sum_{\mathbf{p} \in \mathcal{P}} ||f^\mathbf{w}(g_\mathbf{w}(\mathbf{x}, \mathbf{p})) - \text{rgb}(\mathbf{p}, \mathcal{S}_\mathbf{x})||_1 \tag{1}$$

where $\mathcal{B}$ is a input mini-batch, $\mathcal{P}$ is a sub-sample of points, $\mathbf{w}$ are the neural parameters of encoder and decoder, and $g_\mathbf{w}(\mathbf{x}, \mathbf{p}) := \mathbf{F}_1^\mathbf{w}(\mathbf{p}), \ldots, \mathbf{F}_n^\mathbf{w}(\mathbf{p})$.

*Inference* At inference time we feed the partial textured surface, $\mathcal{T}_i$, and its surface reconstruction (untextured) (Sec. 3.2) as input into the texture inference. The surface reconstruction is a mesh and for all its vertices we regress the rgb value and attach it to the mesh. This yields our final colored surface reconstruction. See Figure 1 for an overview of our method.

### 3.2 Geometry Reconstruction

During inference, we predict the geometry using an IF-Net and condition the texture generation (Sec. 3.1) on this geometry. For learning the geometry reconstruction, we use a standard IF-Net [2]. Our training data consists of pairs $(\mathcal{T}_i, \mathcal{S}_i)_i$, of partial surfaces, $\mathcal{T}_i$, and complete ground truth surfaces, $\mathcal{S}_i$. The shape prediction given $\mathcal{T}_i$, of IF-Nets is done implicitly, by predicting for any point $\mathbf{p}$ in the continuous space $\mathbb{R}^3$, if $\mathbf{p}$ is inside (classification as 1) or outside (classification as 0) of the ground truth $\mathcal{S}_i$. For inference of surfaces, these predictions are evaluated on a dense gird in $\mathbb{R}^3$ and converted to a mesh using marching cubes [5].

*Encoding & decoding* is done with the same architecture described above. However, the input to the encoder consists *only* of one (untextured) channel, created from the partial surface. To encode the surface we again, sample points on it and create a voxel grid. We mark a voxel with 1, if it is a nearest neighbour of a point, and 0 else. We use a grid resolution of $256^3$. Instead of 3 values for rgb we decode into one value for the above mentioned occupancy classification.

*Training* For each ground truth surface, $S_i$, we sample 100.000 points and compute their occupancy. Samples are created in the surface vicinity to concentrate the model capacity on details. To this end, we first uniformly at random sample surface points $\mathbf{p}_{\mathcal{S}}$ and apply Gaussian distributed displacements $\mathbf{n} \sim \mathcal{N}(0, \boldsymbol{\Sigma})$, i.e. $\mathbf{p} := \mathbf{p}_{\mathcal{S}} + \mathbf{n}$. We use a diagonal co-variance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{3\times3}$ with entries $\boldsymbol{\Sigma}_{i,i} = \sigma$. To capture detail we sample 50% of the point samples very near the surface with a small $\sigma_1 = 0.015$, and to learn about the global, more homogeneously filled space, we sample 50% with a larger $\sigma_2 = 0.2$. During training 50.000 points are sub-sampled and learning is unaltered from [2].

## 4 SHARP ECCV 2020 Challenge

In this section we present the challenges and the qualitative results of our method. For a quantitative result analysis, including other participants and baselines, we refer to the SHARP summary paper corresponding to this workshop.

### 4.1 Challenge 1

**Track 1** For track 1 we were kindly given access to the 3DBodyTex.v2 dataset, similar in quality to 3DBodyTex [12][3]. It consists of about 2500 of human scans with high diversity in poses. Some humans wear minimal fitness clothing others are fully clothed in varying clothing. We train our algorithm solely on the officially provided training set. The task is to reconstruct a completed textured 3D surface from a 3D human scan with large missing areas. The ground truth surfaces, $\mathcal{S}_i$, are the raw scans. The textured partial surfaces, $\mathcal{T}_i$, are generated synthetically from the ground truth shapes, using the provided script with default values[4]. For each ground truth mesh we generate 4 partial scans with different incompleteness patterns. The partial data should simulate an acquisition produced by a hand-held 3D scanner. For all scans we found the bounding box with

$$(min_x, max_x, min_y, max_y, , min_z, max_z) = (-0.8, 0.8, -0.15, 2.1, -0.8, 0.8)$$

to be sufficient and use it to place our voxel grid encoding (see Sec. 3.1). In Fig. 2 we show the results obtained with our method from Sec. 3.

[3] https://cvi2.uni.lu/datasets/
[4] https://gitlab.uni.lu/cvi2/eccv2020-sharp-workshop/-/tree/master

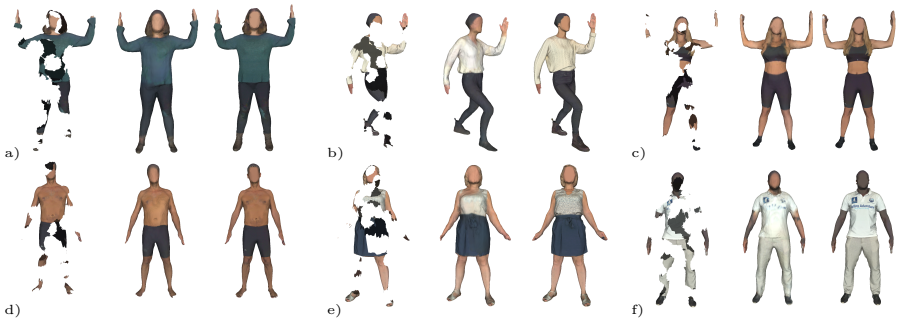


**Fig. 2.** Six testing results a)-f) from our method (unseen during training), left to right: textured, partial input, textured reconstruction, ground truth. Our geometry reconstructions notably show generated body parts in correct positions even for largely missing body parts, see right arm of person e). Our texture generation is visibly coherent with the reconstructed geometry, through the geometry conditioning, see transition from shirt to leggings of b). The texture generation learned can rely on symmetry for generation, see the right foot of person in c).

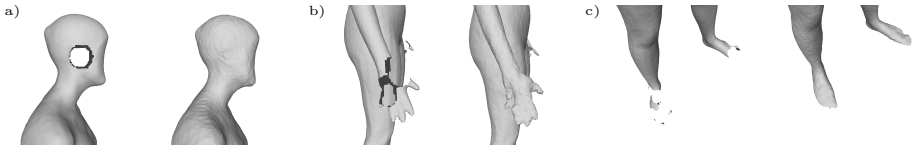


**Fig. 3.** Results on evaluation data (GT disclosed), for reconstruction of the ear region a), hand b) and feet c) regions. Although trained for geometry reconstruction of holes at arbitrary regions, plausible shape completion for these regions is achieved.

**Track 2** In Track 2, the task is to reconstruct fine geometry details of the body such as ears, fingers, feet or a nose, without texture. For this we are given around 1000 detailed ground truth meshes, $\mathcal{S}_i$, obtained by fitting a human body model to human data in minimal clothing from 3DBodyTex.v2. See [13, 14] for methods on automatic and robust model fitting proposed by the dataset authors. On these ground truth meshes a scanning process of the Shapify booth is simulated in software. Additionally, we again shoot holes into the mesh using the provided script in its default setting, shooting holes also at other body parts. Again, we generate 4 partial scans with different incompleteness patterns. Only when the evaluation data was released, we found, that holes only occur at ears, fingers, feet or noses. Therefore, an obvious enhancement for our training would be to concentrate model capacity to those regions by only shooting holes there. We reuse the bounding box found for Track 1. In Fig. 3 we show qualiatative results obtained with our method, IF-Nets.

**Fig. 4.** Results on challenge evaluation data (GT disclosed), for reconstruction of textured, arbitrary objects. This is a highly challenging dataset and problem: objects come from no common class. Thus, no shape prior of objects can be learned - the textured reconstruction can solely rely on surface statistics common in any object. Our network performs reasonable textured reconstruction of what we believe to be a head, orange juice, a shoe and an Egyptian statue.

### 4.2 Challenge 2

In this challenge we are given a dataset (3DObjectTex.v1) of 2000, high quality, textured 3D ground truth scans, $\mathcal{S}_i$, of very diverse objects (real and stuffed animals, statues, technology, furniture,...). When scanning with a hand-held 3D scanner, technical infeasibility of reaching some parts of an object can lead to partial textured scans. Therefore, the task of this challenge is to reconstruct a textured, complete mesh from such partial data. In contrast to challenge 1, the diversity of objects makes it infeasible to build object specify shape priors. As before, we create 4 partial scans per ground truth. Due to huge size variations, we rescale all objects to lie in [-0.5,0.5] in each dimension, by subtracting each vertex with the object center and multiplying by the inverse of its longest bounding box edge. In Fig. 4 we show qualitative results obtained with our IF-Nets.

## 5 Discussion and Conclusions

We proposed an extension of IF-Nets for *textured*, complete 3D reconstruction from textured, highly incomplete 3D scans. For this we predict a continuous texture field, that is, we predict the rgb color at any given point in 3D. Our model won the SHARP ECCV'20 challenge, achieving highest performance on all challenges.

We found that 3D texture completion benefits from incorporating local and global deep features extracted from *both* the 3D partial texture and completed geometry. Specifically, given the partial 3D texture and the 3D geometry completed with IF-Nets, our model successfully in-paints plausible texture in consistence with the completed geometry. We hypothesize, this is because areas of homogeneous texture are often strongly correlated with the geometry: a uni-colored t-shirt or dress can be well textured given a geometry segmentation.

Future work can address to generate higher frequency patterns of texture currently missing in the completions. A promising direction is to use our 3D texture completions, in a coarse to fine framework, as a conditioning for high frequency generative models in 2D.

## References


1. Chen, Z., Zhang, H.: Learning implicit fields for generative shape modeling. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019. pp. 5939–5948 (2019)
2. Chibane, J., Alldieck, T., Pons-Moll, G.: Implicit functions in feature space for 3d shape reconstruction and completion. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE (jun 2020)
3. Genova, K., Cole, F., Sud, A., Sarna, A., Funkhouser, T.: Deep structured implicit functions. In: 2020 IEEE Conference on Computer Vision and Pattern Recognition (2019)
4. Jiang, C.M., Sud, A., Makadia, A., Huang, J., Nießner, M., Funkhouser, T.: Local implicit grid representations for 3d scenes. In: 2020 IEEE Conference on Computer Vision and Pattern Recognition (2020)
5. Lorensen, W.E., Cline, H.E.: Marching cubes: A high resolution 3d surface construction algorithm. In: Computer Graphics and Interactive Techniques. pp. 163–169 (1987). https://doi.org/10.1145/37401.37422, https://doi.org/10.1145/37401.37422
6. Mescheder, L.M., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A.: Occupancy networks: Learning 3d reconstruction in function space. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019. pp. 4460–4470 (2019)
7. Michalkiewicz, M., Pontes, J.K., Jack, D., Baktashmotlagh, M., Eriksson, A.P.: Deep level sets: Implicit surface representations for 3d shape inference. CoRR **abs/1901.06802** (2019), http://arxiv.org/abs/1901.06802
8. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. arXiv preprint arXiv:2003.08934 (2020)
9. Niemeyer, M., Mescheder, L., Oechsle, M., Geiger, A.: Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In: Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2020)
10. Oechsle, M., Mescheder, L., Niemeyer, M., Strauss, T., Geiger, A.: Texture fields: Learning texture representations in function space. In: International Conference on Computer Vision (Oct 2019)
11. Park, J.J., Florence, P., Straub, J., Newcombe, R.A., Lovegrove, S.: Deepsdf: Learning continuous signed distance functions for shape representation. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019. pp. 165–174 (2019)
12. Saint, A., Ahmed, E., Shabayek, A.E.R., Cherenkova, K., Gusev, G., Aouada, D., Ottersten, B.: 3dbodytex: Textured 3d body dataset. In: 2018 International Conference on 3D Vision (3DV). pp. 495–504 (2018)
13. Saint, A., Rahman Shabayek, A.E., Cherenkova, K., Gusev, G., Aouada, D., Ottersten, B.: Bodyfitr: Robust automatic 3d human body fitting. In: 2019 IEEE International Conference on Image Processing (ICIP). pp. 484–488 (2019)
14. Saint, A., Shabayek, A., Aouada, D., Ottersten, B., Cherenkova, K., Gusev, G.: Towards automatic human body model fitting to a 3d scan. In: 8th International Conference and Exhibition on 3D Body Scanning and Processing Technologies, Montreal QC, Canada. pp. 274–280 (10 2017). https://doi.org/10.15221/17.274
15. Saito, S., , Huang, Z., Natsume, R., Morishima, S., Kanazawa, A., Li, H.: Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. arXiv preprint arXiv:1905.05172 (2019)

16. Sitzmann, V., Zollhöfer, M., Wetzstein, G.: Scene representation networks: Continuous 3d-structure-aware neural scene representations. In: Advances in Neural Information Processing Systems. pp. 1119–1130 (2019)