Supplementary Material

NSF: Neural Surface Fields for Human Modeling from Monocular Depth

Abstract

In this supplementary document, we provide further details about the network, training procedure, and method. We also report further results on per-outfit reconstruction, unseen subject adaptation, multi-resolution, and real data. Finally, we report timings for the method inference.

1. NSF: Implementation Details

1.1. Network Architecture

Implicit Fusion Shape For learning the implicit fusion shape \( f_{\text{shape}} \), we follow the Auto-Decoder network architecture proposed in IGR [13], as described in sections A.1. We depict it in Figure 1.

![Figure 1: Auto-Decoder architecture proposed in IGR [13].](image)

Global Pose Feature Given the pose parameter \( \theta \) in \( SO(3) \), we use an MLP to extract the global pose features. Here, FC represents a fully connected layer, and FC+ has an additional ReLU() layer and dropout layer with \( p = 0.3 \) after each FC layer. This network extracts 24-dimensional global pose feature from 72-dimensional pose parameters \( \theta \). We visualize this in Figure 2.

![Figure 2: MLP to extract the global pose features.](image)

Neural Surface Field Our proposed neural surface field predicts the continuous pose-dependent displacement field based on the subject-specific fusion shape with the Auto-Decoder network reported in Figure 3.

![Figure 3: Auto-Decoder network for Neural Surface Field.](image)

FC indicates fully connected layer. FC+ contains a FC layer, a batch normalization layer, and a softplus activation layer. \( F^* (x^c) \) indicates the local feature at \( x \) on the feature surface, which has the dimension of 64.

1.2. Training Details

The whole model is trained on a single Nvidia 2080-Ti GPU with PyTorch deep learning framework [26]. We use two separate ADAM optimizers [15] for network parameters and the feature space. For both optimizers we use initial learning rate of \( 1 \times 10^{-4} \) scheduled to decrease by a factor of 2 every 500 epochs. At the fine-tuning stage, we freeze network parameters and only optimize the feature space.
1.3. Evaluation Metrics

**Chamfer Distance (CD)** We report the bi-directional square-root Chamfer distance (CD), which measures L2 distances between the reconstructed surface to the ground-truth surface (lower is better). Given the reconstructed shape and the ground-truth shape, we sample with 200,000 points \( X \) and \( Y \) from them. We calculate the root mean square bi-directional Chamfer distance with:

\[
\left( \frac{1}{2M} \sum_{i=1}^{M} \min_j \|x_i - y_j\|_2^2 + \frac{1}{2N} \sum_{j=1}^{N} \min_i \|x_i - y_j\|_2^2 \right)^{\frac{1}{2}},
\]

where \( M \), \( N \) are the number of points from the sampled point cloud \( X \) and \( Y \), respectively. The unit of reported result in the paper is centimeter (cm).

**Normal Consistency (NC)** We also report the normal consistency (NC), which measures the accuracy and the completeness of reconstructed shape normals (higher is better). Following the notion in Eq. 1, we compute the bi-directional normal consistency as

\[
\frac{1}{2M} \sum_{i=1}^{M} \frac{n(x_i)}{\|n(x_i)\|} \frac{n(y_i)}{\|n(y_i)\|} + \frac{1}{2N} \sum_{j=1}^{N} \frac{n(x_j)}{\|n(x_j)\|} \frac{n(y_j)}{\|n(y_j)\|},
\]

where

\[
\hat{y} = \arg \min_{y, \in Y} d(x_i, y_j),
\]

\[
\hat{x} = \arg \min_{x, \in X} d(x_i, y_j),
\]

\( n(\cdot) \) denotes the normal of the given point and \( d(x, y) \) the \( L_2 \) distance between point \( x \) and \( y \).

**Intersection Over Union (IoU)** Volumetric intersection over union (IoU) measures how well the reconstructed mesh matches the ground-truth mesh (higher is better). Following IF-Nets [7], we perform the implicit waterproofing of ground-truth scan. We sample 200,000 points \( X \) in the bounding box of the ground-truth scan and query the occupancy status of each sampled points in reconstructed shape and the ground-truth shape. We compute the IoU with:

\[
\frac{\text{occ}_{\text{pred}}(X) \cap \text{occ}_{\text{gt}}(X)}{\text{occ}_{\text{pred}}(X) \cup \text{occ}_{\text{gt}}(X)},
\]

where \( \text{occ}_{\text{pred}} \) is the occupancy query function w.r.t. predicted shape and \( \text{occ}_{\text{gt}} \) is the occupancy query function w.r.t. ground-truth shape.

1.4. Baselines

Our baselines are PINA [10] and DSFN [4], which both model clothed humans from monocular depth input. Although the authors promised the code in their work, it is not yet available at the time of writing. Hence, we perform a quantitative comparison using results from respective original papers and a qualitative comparison using meshes provided by them. Moreover, we apply two naive baselines, namely SMPL [17] and our fusion shape. The fusion shape can be understood as SMPL+D. It shows how our proposed NSF learns the pose-dependent deformation.

1.5. SMPL Fitting

For the shapes without given SMPL registration (e.g. BuFF [33] for synthetic data as well as DSFN [4] for real data), we use a concurrent work based on LVD [8] to fit the SMPL [17] model to the input point cloud. Moreover, we use off-the-shelf methods [11, 31, 32] to estimate the SMPL [17] parameters from RGB images. Since the estimation from RGB images is always inaccurate in terms of root rotation and leg poses, we use the RVH registration library [1, 2, 3] to manually refine the SMPL fit with the input point cloud. Finally, we choose the visually plausible SMPL fit given the input point cloud.

2. NSF: Method Details

2.1. Extract Fusion Shape

Here we provide more detail in the learning of the canonical fusion shape. For the outfit which can be described by SMPL topology, we additionally fit SMPL+D using our learned implicit fusion shape surface with:

\[
\min_{\phi} f_{\text{shape}}(X(\text{SMPL+D})|\phi_{\text{shape}}),
\]

where \( X(\text{SMPL+D}) \) denotes the sampled points from the SMPL+D shape, \( f_{\text{shape}} \) together with \( \phi_{\text{shape}} \) indicates the implicit fusion shape of the subject with outfit. We push the sampled points lie on the zero-level set to obtain the displacement of the SMPL surface. To ensure the smoothness of the SMPL+D surface, we additionally minimize the Laplacian of the optimized mesh [25].

Our learned fusion shape defines a coherent surface for our proposed Neural Surface Field (NSF). We can fuse the depth observation and texture from multiple partial shapes onto this unifying surface. This enables us to learn the deformation and the texture in the NSF.

2.2. Texture Transfer on Fusion Shape

To transfer the texture color from the posed partial shapes to the canonical fusion one, first we randomly select 10 partial shapes which could cover the most area of the fusion shape. Then, we canonicalize (Eq.1 in main paper) and project (Eq.2 in main paper) them to find the canonical correspondence on the fusion shape, as shown in Figure 5. Finally, we assign the color of each vertex on the fusion shape from the nearest canonical points. We found that
2.3. Surface-based Feature Query

We use the vertices of our learned fusion shape to form the basis of the surface-based feature space. As introduced in Eq. 6 in the main paper, we can lift the neural surface feature from $x^{cc}$, $F^s(x^c) \leftarrow F^s(x^{cc})$ with the surface projection. To obtain the $F^s(x^{cc})$, we use Kaolin [12] to estimate the barycentric coordinate w.r.t. three vertices of the corresponding triangle and interpolate features of three vertex features using the barycentric weights.

3. Further Results

3.1. Per-Outfit Reconstruction Result

To prove the effectiveness of having a single general decoder for all the outfits, we also train outfit-specific ones and compare ourselves. We report the per-outfit reconstruction quantitative results of our method in Table 1. Our generalizable decoder achieves better reconstruction quality on every single outfit, and so also on average. We remark that our decoder uses $\frac{1}{9}$ of the parameters compared to the per-outfit training strategy.

3.2. Adapt to Unseen Subject

Here we study the method performance using a varying number of frames during the adaptation of surface-based features of unseen subjects. It shows that our method can reconstruct the given partial shape well using the only 10 frames. Besides, the more frames used to fine-tune, the more the model generalizes to the unseen poses.

<table>
<thead>
<tr>
<th>Operation</th>
<th>Seen Frames</th>
<th>Unseen Poses</th>
</tr>
</thead>
<tbody>
<tr>
<td>CD ↓</td>
<td>NC ↑</td>
<td>IoU ↑</td>
</tr>
<tr>
<td>10 Frames</td>
<td>0.75</td>
<td>0.920</td>
</tr>
<tr>
<td>20 Frames</td>
<td>0.75</td>
<td>0.919</td>
</tr>
<tr>
<td>50 Frames</td>
<td>0.77</td>
<td>0.918</td>
</tr>
</tbody>
</table>

Table 2: Our model can be quickly adapted to unseen subjects by only fine-tuning the surface-based feature space with limited number of partial shapes. We use different splits from the one used in the main, thus the quantitative number here varies from Table 3 in main paper.

3.3. Multi-resolution Flexibility

Our proposed approach has high flexibility to generate desired mesh resolution and topology. We represent our fu-
<table>
<thead>
<tr>
<th>Decoder Type</th>
<th># Decoder</th>
<th>Subject</th>
<th>Garment</th>
<th>CAPE Data [21, 27, 33]</th>
<th></th>
<th>Mean Value</th>
<th>Median Value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CD (cm) ↓</td>
<td>NC ↑</td>
<td>IoU ↑</td>
<td>CD (cm) ↓</td>
</tr>
<tr>
<td>Outfit-specific</td>
<td>1</td>
<td>00122</td>
<td>shortlong</td>
<td>0.58</td>
<td>0.941</td>
<td>0.919</td>
<td>0.52</td>
</tr>
<tr>
<td>Generalizable</td>
<td>/</td>
<td>00122</td>
<td>shortlong</td>
<td>0.52</td>
<td>0.952</td>
<td>0.935</td>
<td>0.46</td>
</tr>
<tr>
<td>Outfit-specific</td>
<td>1</td>
<td>00122</td>
<td>shortshort</td>
<td>0.65</td>
<td>0.922</td>
<td>0.906</td>
<td>0.66</td>
</tr>
<tr>
<td>Generalizable</td>
<td>/</td>
<td>00122</td>
<td>shortshort</td>
<td>0.62</td>
<td>0.933</td>
<td>0.915</td>
<td>0.63</td>
</tr>
<tr>
<td>Outfit-specific</td>
<td>1</td>
<td>00215</td>
<td>jerseyshort</td>
<td>0.49</td>
<td>0.949</td>
<td>0.934</td>
<td>0.49</td>
</tr>
<tr>
<td>Generalizable</td>
<td>/</td>
<td>00215</td>
<td>jerseyshort</td>
<td>0.46</td>
<td>0.956</td>
<td>0.944</td>
<td>0.47</td>
</tr>
<tr>
<td>Outfit-specific</td>
<td>1</td>
<td>00215</td>
<td>poloshort</td>
<td>0.70</td>
<td>0.935</td>
<td>0.904</td>
<td>0.65</td>
</tr>
<tr>
<td>Generalizable</td>
<td>/</td>
<td>00215</td>
<td>poloshort</td>
<td>0.62</td>
<td>0.945</td>
<td>0.920</td>
<td>0.56</td>
</tr>
<tr>
<td>Outfit-specific</td>
<td>1</td>
<td>00215</td>
<td>shortlong</td>
<td>0.63</td>
<td>0.927</td>
<td>0.910</td>
<td>0.60</td>
</tr>
<tr>
<td>Generalizable</td>
<td>/</td>
<td>00215</td>
<td>shortlong</td>
<td>0.58</td>
<td>0.941</td>
<td>0.925</td>
<td>0.57</td>
</tr>
<tr>
<td>Outfit-specific</td>
<td>1</td>
<td>03375</td>
<td>blazerlong</td>
<td>0.83</td>
<td>0.897</td>
<td>0.882</td>
<td>0.79</td>
</tr>
<tr>
<td>Generalizable</td>
<td>/</td>
<td>03375</td>
<td>blazerlong</td>
<td>0.77</td>
<td>0.926</td>
<td>0.894</td>
<td>0.73</td>
</tr>
<tr>
<td>Outfit-specific</td>
<td>1</td>
<td>03375</td>
<td>longlong</td>
<td>0.86</td>
<td>0.899</td>
<td>0.872</td>
<td>0.76</td>
</tr>
<tr>
<td>Generalizable</td>
<td>/</td>
<td>03375</td>
<td>longlong</td>
<td>0.79</td>
<td>0.931</td>
<td>0.890</td>
<td>0.71</td>
</tr>
<tr>
<td>Outfit-specific</td>
<td>1</td>
<td>03375</td>
<td>shortlong</td>
<td>0.81</td>
<td>0.919</td>
<td>0.878</td>
<td>0.76</td>
</tr>
<tr>
<td>Generalizable</td>
<td>/</td>
<td>03375</td>
<td>shortlong</td>
<td>0.72</td>
<td>0.939</td>
<td>0.898</td>
<td>0.66</td>
</tr>
<tr>
<td>Outfit-specific</td>
<td>1</td>
<td>03375</td>
<td>shortshort</td>
<td>0.86</td>
<td>0.907</td>
<td>0.871</td>
<td>0.83</td>
</tr>
<tr>
<td>Generalizable</td>
<td>/</td>
<td>03375</td>
<td>shortshort</td>
<td>0.79</td>
<td>0.933</td>
<td>0.887</td>
<td>0.77</td>
</tr>
<tr>
<td><strong>Ours, Outfit-specific</strong></td>
<td>9</td>
<td>3 subjects</td>
<td>9 outfits</td>
<td>0.71</td>
<td>0.922</td>
<td>0.897</td>
<td>0.67</td>
</tr>
<tr>
<td><strong>Ours generalizable</strong></td>
<td>1</td>
<td>3 subjects</td>
<td>9 outfits</td>
<td>0.65</td>
<td>0.940</td>
<td>0.912</td>
<td>0.62</td>
</tr>
</tbody>
</table>

Table 1: We evaluate our method on the task of reconstructing 3D shapes from monocular depth point clouds on CAPE [21, 27, 33] data. We compare the outfit-specific decoders against our general one for all outfits. We appreciate the better performance of our choice in all the settings and metrics.

3.4. Real Data Experiments

We show more qualitative reconstruction result of real data in DSFN [4] dataset in figure 9.

3.5. Inference Time Per Frame

We use the edge connectivity of our extracted fusion shape on the deformed vertices with NSF $V_p$ (see sec. 3.5 in the main paper) to generate the posed mesh. Our formulation is much more efficient compared to other works since we get rid of per-frame Marching cube [18] as in [5, 6, 9, 23, 24, 29, 30] or per-frame Poisson reconstruction [14] as in [16, 19, 20, 22]. X-Avatar [28], a concurrent work based on SNARF [6], reports 7 seconds per frame as inference time which include remeshing via Marching cube on Nvidia RTX6000. In our proposed method, we report an average time of 0.039 seconds per frame, including deformation prediction, Linear Blend Skinning, and remeshing on RTX3080-Ti, which is $\sim 180$ times faster than X-Avatar [28]. Furthermore, we compare the inference time with point-based works POP [22], which requires Poisson reconstruction to obtain mesh from point cloud for each frame. In the same setting, POP requires about 1.6 seconds to reconstruct the surface at each frame. Our approach is $\sim 40$ times faster.
Figure 6: We show the textured fusion shape with different resolution and topology: (a) Marching cube with resolution of 128, ~ 5k vertices; (b) SMPL+D topology, ~ 7k vertices; (c) subdivided SMPL+D topology, ~ 27k vertices; (d) Marching cube with resolution of 512, ~ 86k vertices.

Figure 7: Learned fusion shapes of subjects from BuFF \cite{33, 27} (1st row), CAPE \cite{21, 27} (2nd row), DSFN \cite{4} (left on 3rd row), Resynth \cite{22, 20} (3 right on 3rd row).

References

Figure 8: Given the partial shape (a), we show the reconstruction with different resolution and topology with our NSF: (b) Marching cube with resolution of 128, $\sim 5k$ vertices; (c) SMPL+D topology, $\sim 7k$ vertices; (d) subdivided SMPL+D topology, $\sim 27k$ vertices; (e) Marching cube with resolution of 512, $\sim 88k$ vertices.
Figure 9: Our reconstruction result on Real Data from DSFN [4] dataset. Note that our method only takes noisy depth images as input. Our approach shows robustness while still capturing finer details of the input (e.g., wrinkles).


