Supplementary Materials for 3DFG-PIFu: 3D Feature Grids for Human Digitization from Sparse Views

Kennard Yanting Chan[1,2], Fayao Liu[2], Guosheng Lin[1], Chuan Sheng Foo[2,3], and Weisi Lin[1]

[1] Nanyang Technological University, Singapore
[2] Institute for Infocomm Research, A*STAR
[3] Centre for Frontier AI Research, A*STAR

1 Comparison of our Method with additional SOTA methods

1.1 Comparison of our Method with the DoubleField

Besides DeepMultiCap [10], DoubleField [9] is another SOTA method that uses self-attention mechanism to combine features from different views together. In this section, we will compare our 3DFG-PIFu model against DoubleField.

We trained DoubleField with the same training data (THuman2.0 dataset) that we have trained our 3DFG-PIFu model on, and we show the results in Fig. 1. We find that, compared to our 3DFG-PIFu model, the meshes produced by DoubleField suffer from structural inaccuracies and lack of appearance details. The meshes shown in the figure are some of the best ones that DoubleField has produced. The results of DoubleField differ from our expectations, and one reason could be that DoubleField requires more training meshes and training meshes of a higher quality. The authors of DoubleField trained their model on 1500 human meshes collected from Twindom[4]. Twindom is not open-source, and we are unable to inspect the quality of these meshes. But 1500 human meshes is around thrice the size of our training data.

1.2 Comparison of our Method with the 'Data-Driven 3D Reconstruction' method by Zins et al.

Another existing pixel-aligned implicit method that uses multiple views to form a 3D clothed human mesh is 'Data-Driven 3D Reconstruction', and it is proposed by Zins et al. in [11]. The 'Data-Driven 3D Reconstruction' method also uses self-attention mechanism to combine features from different views together.

A qualitative evaluation between our 3DFG-PIFu models and the 'Data-Driven 3D Reconstruction' method is presented in Fig. 2. Column (c) shows meshes produced by a 3DFG-PIFu model that is given a 512x512 RGB image

[4] https://web.twindom.com/

as input. Column (d) shows meshes produced by a 3DFG-PIFu model that is given a 1024x1024 RGB image as input. Neither of the two 3DFG-PIFu models is given groundtruth SMPL-X meshes as input. The figure shows that our models clearly outperformed the 'Data-Driven 3D Reconstruction' method.

In addition, we provided a quantitative evaluation in Tab. 1. The table shows that our models significantly outperformed the 'Data-Driven 3D Reconstruction' method.

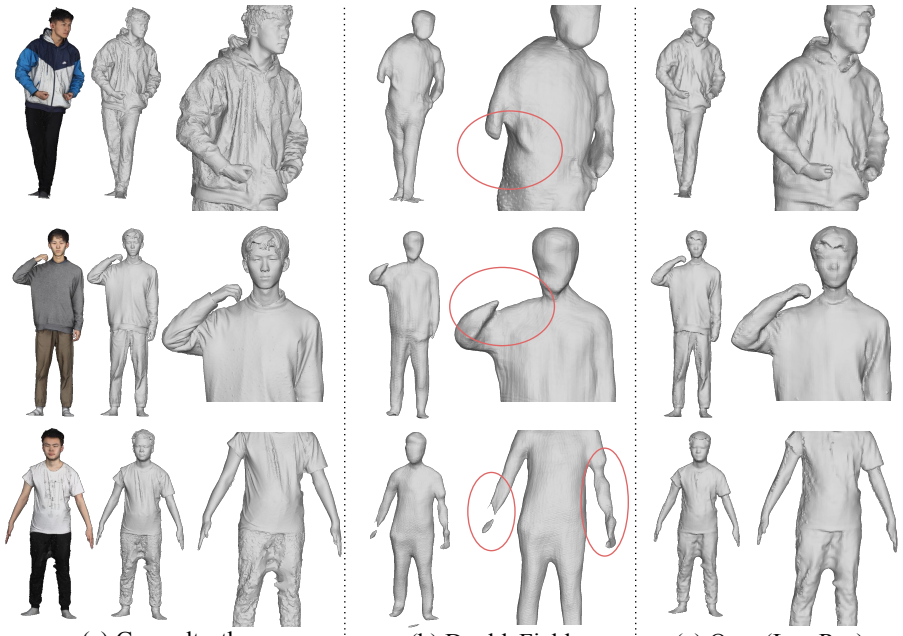

Fig. 1: DoubleField produces meshes with structural issues.

Table 1: Quantitative evaluation of our models against the 'Data-Driven 3D Reconstruction' method [11] ('HR' indicates if a 1024x1024 RGB image is used in the method. By default, a 512x512 RGB image is used.)

		THuman2.0		BUFF		MultiHuman	
Methods	HR	CD (10^{-5})	P2S (10^{-5})	CD (10^{2})	P2S (10^{2})	CD (10^{-5})	P2S (10^{-5})
Data-Driven 3D Reconstruction	×	12.67	21.37	4.158	7.340	8.812	11.63
Ours (No HR, No GT Smplx)	×	5.796	5.811	2.509	2.286	6.320	5.737
Ours (HR, No GT Smplx)	✓	**5.133**	**5.028**	**2.508**	**2.121**	**5.315**	**4.866**

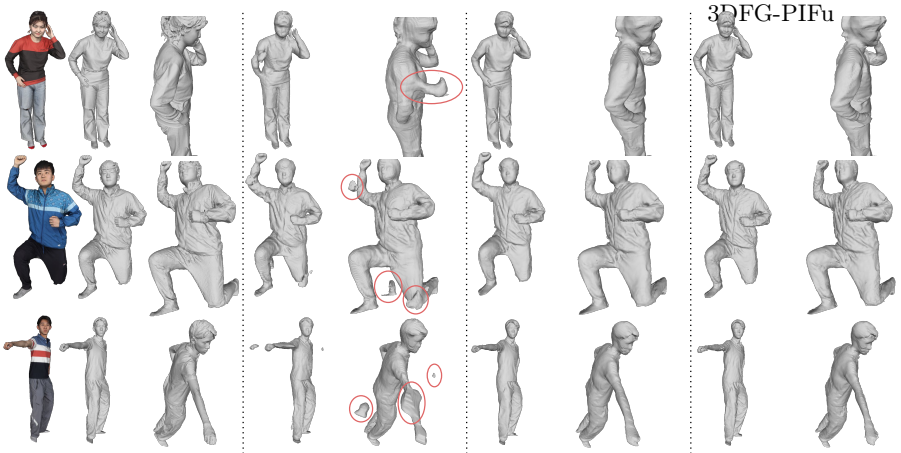

Fig. 2: Qualitative evaluation of our models against an additional SOTA method ('Data-Driven 3D Reconstruction' method by Zins et al. [11]). None of the models here uses groundtruth SMPL-X meshes as inputs.

2 More Analysis of Quantitative Results from the Main Paper

2.1 Analysis of Results from Models using SMPL-X priors

In our main paper, there are both existing SOTA models and our own models that use priors derived from a SMPL-X mesh as an input. However, as seen from our quantitative results in the main paper ('Table 1' in our main paper), the use of priors from a groundtruth SMPL-X mesh may not always lead to better results. This is because even the groundtruth SMPL-X mesh can have pose and shape errors. These errors arise because it is extremely difficult for a clothless and hairless SMPL-X mesh to perfectly fit the corresponding groundtruth clothed human body mesh. For example, a groundtruth clothed human body mesh of a human subject wearing a thick jacket is often given a 'groundtruth' SMPL-X mesh with an unusually large waistline or stomach. These errors diminish the value of using a SMPL-X prior in a model.

However, in many contexts, there is still value in using SMPL-X meshes as priors. In our experimental set-up, we set the angle between each pair of given views to be 90 degree. If we have set the angle to be much smaller, say 10 degree, then the benefit of using SMPL-X meshes will be much more obvious and significant. This is because when the angle is smaller, the depth ambiguity problem becomes more of an issue. This means that the model is unable to tell whether the clothed human body mesh should be located nearer or further away from the camera. Taking into account that weak perspective projection is used to render the images (or views), the model can really only guess the z-position (or distance away from camera) of the clothed human body mesh. But if SMPL-X priors are used, then the depth ambiguity problem can be largely

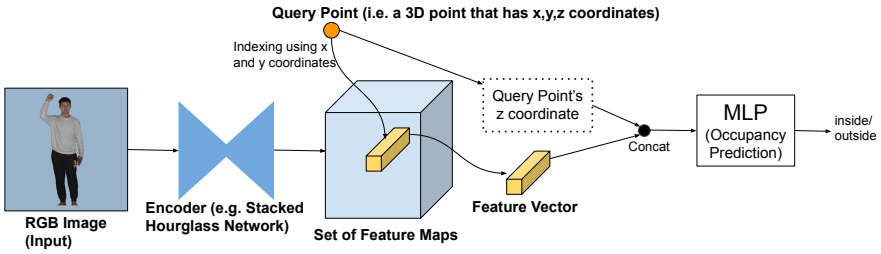

Fig. 3: Pipeline and Architecture of the Single-view PIFu [8] method.

resolved. In other words, if the angle between the pair of views is smaller (e.g. 10 degree), then models that use SMPL-X priors (e.g. SDF-based SMPL-X features, PaMIR's voxel-aligned features, or S-PIFu features) are likely to have a much better results compared models that do not use any SMPL-X priors.

3 Elaborating on the Problems with Existing Methods

Single-view PIFu To fully understand the problems with existing methods for 3D clothed human reconstruction from sparse views, we have to first look at how a generic pixel-aligned implicit model works. Hence, we will look at one of the earliest (and most general) pixel-aligned implicit models, which is the single-view PIFu [8].

In Fig. 3, we show the pipeline of the single-view PIFu method. First, an input image is fed into an Encoder, which is a 2D CNN (e.g. Stacked Hourglass Network). The Encoder outputs a set of feature maps which, as a whole, has the dimensions of (C, H, W), where C = Channels, H = Height, and W = Width. Please note that, for simplicity, we are ignoring the batch dimension here.

This set of feature maps is actually designed to correspond to the 3D camera space of the input image. During training or testing, a set of 3D points (i.e. x,y,z coordinates) are sampled from the 3D camera space of the input image. These 3D points are called either sample points or query points. For each query point, we will take its (x,y) coordinates to index the set of feature maps (as shown in Fig. 3). After indexing, each query point will give us a feature vector. This feature vector is concatenated with the z coordinate of that query point and then fed into a Multilayer Perceptron (MLP). The MLP will then predict the occupancy of that query point (i.e. whether the query point is inside or outside a groundtruth human mesh that has been transformed into the camera space of the input image). In practice, the occupancy prediction outputted by the MLP is a continuous value that ranges from 0 to 1 where the value of 0.5 is interpreted as the surface of a mesh. Any value from 0 to 0.5 is interpreted as being outside the mesh. Any value from 0.5 to 1 is interpreted as being inside the mesh.

During testing, a 3D grid (e.g. 256x256x256) of query points will be passed into the PIFu's pipeline to yield a 3D grid of occupancy predictions. From these occupancy predictions, we will apply the Marching Cubes algorithm [6] to obtain a predicted human mesh.

Multi-view PIFu With the understanding of single-view PIFu, we can now take a closer look at Multi-view PIFu [8]. In this section, we consider the number of views given to a Multi-view PIFu to be 2. Multi-view PIFu is basically an extension of single-view PIFu, and we show the pipeline of Multi-view PIFu in Fig. 4. The first difference between single-view PIFu and Multi-view PIFu is that we have a 3D world coordinate space. The 3D world coordinate space is a space that connects the 3D camera space of view 1 with the 3D camera space of view 2. In other words, any point in the 3D world coordinate space can be transformed into a point in each of the 3D camera spaces, and any point in either of the 3D camera spaces can be transformed into a point in the 3D world coordinate space.

In Multi-view PIFu, as shown in the figure, view 1 and view 2 are separately processed by the same Encoder to generate two different sets of feature maps. Each set of feature maps corresponds to the 3D camera space of a view. Next, query points are sampled from the 3D world coordinate space. Each of these query points are transformed into both the 3D camera spaces of view 1 and view 2. In other words, each query point in the 3D world coordinate space will generate a pair of query points: One query point in the 3D camera space of view 1 and one query point in the 3D camera space of view 2.

As illustrated in the figure, the pair of query points will be used to index their respective set of feature maps, and this will give us two different feature vectors (one from each view). To combine the feature vectors from the two views, often either averaging (used by Multi-view PIFu [8]) or self-attention mechanism (used by DeepMultiCap, DoubleField, and the 'Data-Driven 3D Reconstruction' method) is used. To be more precise, Multi-view PIFu does not average up the two feature vectors but first pass the two feature vectors separately into the MLP. An intermediate layer in the MLP will produce two different feature embeddings from the two feature vectors. The feature embeddings are averaged up to form an average feature embedding, which is then passed into the subsequent layers of the MLP to produce an occupancy prediction for that query point in 3D world coordinate (in green in the figure).

On the other hand, DeepMultiCap, DoubleField, and the 'Data-Driven 3D Reconstruction' method use self-attention (i.e. weighted averaging) to combine the two feature vectors (in orange and red in the figure) into a single feature vector (in purple in the figure). This single feature vector is used by the MLP to produce an occupancy prediction. Now we will explain the problem with existing methods like Multi-view PIFu, DeepMultiCap, DoubleField, and the 'Data-Driven 3D Reconstruction' method.

3.1 The Problem with Combining features of different views only late in the Pipeline

From Fig. 4, we see that existing methods only combine features of different views **after** the query points in different camera spaces are used to index the sets of feature maps. In other words, the features of different views are combined only locally and not globally. This happens because the two sets of feature maps are not directly combined together. Each time before the MLP makes a prediction,

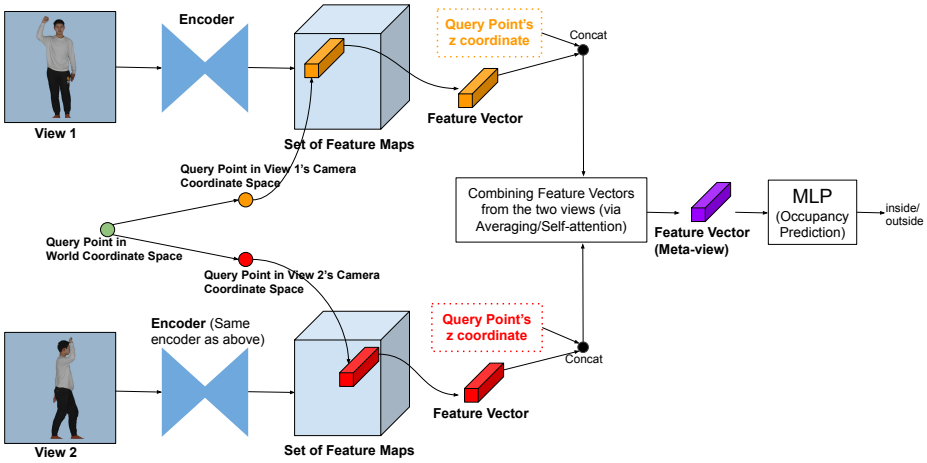

Fig. 4: Pipeline and Architecture of the Multi-view PIFu [8] method. Assume the number of views given is 2.

only a selected part (a feature vector) from the two sets of feature maps are combined together. The resulting problem is that the two views are not combined in a cohesive and coherent manner.

3.2 The Problem with using Averaging or Self-attention mechanism to combine features of different views

Using Fig. 4, we see that it is possible for a query point in world coordinate space to be visible in view 1 but not visible in view 2. In such a case, more weight should be given to the feature vector (corresponding to that query point) from view 1 and less weight should be given to the corresponding feature vector from view 2. As such, taking the average of the two feature vectors (or the two feature embeddings) is not optimal.

As for self-attention mechanism, we can see from Fig. 4 that the two Sets of Feature Maps are each generated using information from one view and without any information from the other view. With a query point in world coordinate space P_w, two feature vectors (each from a Set of Feature Maps) can be obtained as aforementioned. The problem arises when the two feature vectors disagree with each other. The feature vector from the first Set of Feature Maps may believe that P_w lies outside the human body mesh. But the feature vector from the second Set of Feature Maps may believe otherwise. The self-attention mechanism works by acting on (and analysing) the two feature vectors in a bid to obtain a single, meta-view feature vector. In this case, the self-attention mechanism will not perform well because both of two feature vectors believe that they are correct, and the self-attention mechanism only has access to information in these two feature vectors. Thus, a coherent meta-view feature vector cannot be produced by the self-attention mechanism.

Table 2: Efficiency Comparison against SOTA methods. 'SM' = SMPL-X mesh is used. 'HR' = 1024x1024 RGB images are used. By default, 512x512 RGB images are used.

Methods	Time per test instance (ms)
Multi-view PIFu	**1437**
IntegratedPIFu	1564
DeepMultiCap	9817
'Data-Driven 3D Reconstruction' by Zins et al.	6798
DoubleField	6311
Ours (No HR, No SM)	**1759**
Ours (HR, No SM)	1796
Ours (No HR, w SM)	1886
Ours (HR, w SM)	1927

SeSDF [1] attempted to address this by using a SMPL-X mesh to identify which of the two feature vectors is more visible and assign a greater weight to the more visible feature vector. But a SMPL-X mesh is a clothless and hairless representation of a human body. This coarse and imprecise representation will inherently have errors in deciding which feature vectors are more visible. Moreover, it is impractical to assume that we will always have access to the groundtruth SMPL-X mesh. A non-groundtruth (i.e. predicted) SMPL-X mesh will inevitably have substantial pose and shape errors, which make it even more difficult to use the SMPL-X mesh for deciding which of the two feature vectors is more visible.

4 Efficiency and Computation Time of 3DFG-PIFu

While efficiency is not a claim made by our 3DFG-PIFu, we understand concerns related to it. 3DFG-PIFu's efficiency can be drastically improved by some implementation tricks that we developed. For example, instead of using the Marching Cubes algorithm multiple times (for base mesh, partial refined meshes, and final mesh), we do it only for the final mesh. This is possible by keeping the base mesh and partial refined meshes in their SDF form (i.e. a 3D Feature Grid of SDF values). Using these 3D Feature Grids, we are able to obtain required features such as G'_S and G_V using hash maps and optimised Numpy functions. Given that Marching Cubes takes the bulk of the inference time, this implementation trick made 3DFG-PIFu very efficient.

As seen in Tab. 2, compared to SOTA methods, 3DFG-PIFu fared reasonably well in terms of efficiency. Values reported are the average test time with a NVIDIA RTX A5000 GPU. For our models, we included time incurred to generate our required features like G_N, G_M, G_X, G'_S, and G_V.

5 Training details and Hardware used

3DFG-PIFu is trained with a RMSprop optimizer with an initial learning rate of 10^{-3}. The encoders (in both the 1st stage and 2nd stage) are each a stacked

hourglass network [7] that consists of 4 stacks. Our Multilayer Perceptron (MLP) has layers with the following dimensions: (257, 1024, 512, 256, 128, 1). The 3rd, 4th, 5th layers of our MLP are fitted with skip connections (same as [2–5]). During training, 8000 sample points (or query points) are used for every RGB image that is fed to the model. More details about the other hyperparameters can be found in our source code, which will be released publicly. These hyperparameter values are chosen using grid search.

In terms of hardware, we train all our models using NVIDIA RTX A5000 GPUs. Minimally, 1 GPU is required to run our training script.

References

1. Cao, Y., Han, K., Wong, K.Y.K.: Sesdf: Self-evolved signed distance field for implicit 3d clothed human reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4647–4657 (2023)
2. Chan, K., Lin, G., Zhao, H., Lin, W.: S-pifu: Integrating parametric human models with pifu for single-view clothed human reconstruction. Advances in Neural Information Processing Systems **35**, 17373–17385 (2022)
3. Chan, K.Y., Lin, G., Zhao, H., Lin, W.: Integratedpifu: Integrated pixel aligned implicit function for single-view human reconstruction. In: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part II. pp. 328–344. Springer (2022)
4. Chan, K.Y., Liu, F., Lin, G., Foo, C.S., Lin, W.: Fine structure-aware sampling: A new sampling training scheme for pixel-aligned implicit models in single-view human reconstruction. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 964–971 (2024)
5. Chan, K.Y., Liu, F., Lin, G., Foo, C.S., Lin, W.: R-cyclic diffuser: Reductive and cyclic latent diffusion for 3d clothed human digitalization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10304–10313 (2024)
6. Lorensen, W.E., Cline, H.E.: Marching cubes: A high resolution 3d surface construction algorithm. ACM siggraph computer graphics **21**(4), 163–169 (1987)
7. Newell, A., Yang, K., Deng, J.: Stacked hourglass networks for human pose estimation. In: European conference on computer vision. pp. 483–499. Springer (2016)
8. Saito, S., Huang, Z., Natsume, R., Morishima, S., Kanazawa, A., Li, H.: Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2304–2314 (2019)
9. Shao, R., Zhang, H., Zhang, H., Chen, M., Cao, Y.P., Yu, T., Liu, Y.: Doublefield: Bridging the neural surface and radiance fields for high-fidelity human reconstruction and rendering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15872–15882 (2022)
10. Zheng, Y., Shao, R., Zhang, Y., Yu, T., Zheng, Z., Dai, Q., Liu, Y.: Deepmulticap: Performance capture of multiple characters using sparse multiview cameras. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6239–6249 (2021)
11. Zins, P., Xu, Y., Boyer, E., Wuhrer, S., Tung, T.: Data-driven 3d reconstruction of dressed humans from sparse views. In: 2021 International Conference on 3D Vision (3DV). pp. 494–504. IEEE (2021)