DynoSurf: Neural Deformation-based Temporally Consistent Dynamic Surface Reconstruction (Supplementary)

Yuxin Yao[1], Siyu Ren[1], Junhui Hou[1]⋆, Zhi Deng[2], Juyong Zhang[3], and Wenping Wang[4]

[1]City University of Hong Kong [2]Tencent Games
[3]University of Science and Technology of China [4]Texas A&M University

1 Supplement of ablation study

In this section, we further discuss the effectiveness of some components of our method, as well as the robustness of the input data.

Keyframe selection and template surface learning. Compared to directly setting the middle frame as a keyframe, Fig. 1 shows that the selected keyframe by our strategy is closer to the average shape. After learning the template surface, we also show the results of only using the fine stage (Fig. 1 (b)) and using the coarse-to-fine stage (Fig. 1 (c)). Based on Fig. 1 (c), we further show the enhanced template surface after the temporal reconstruction process.

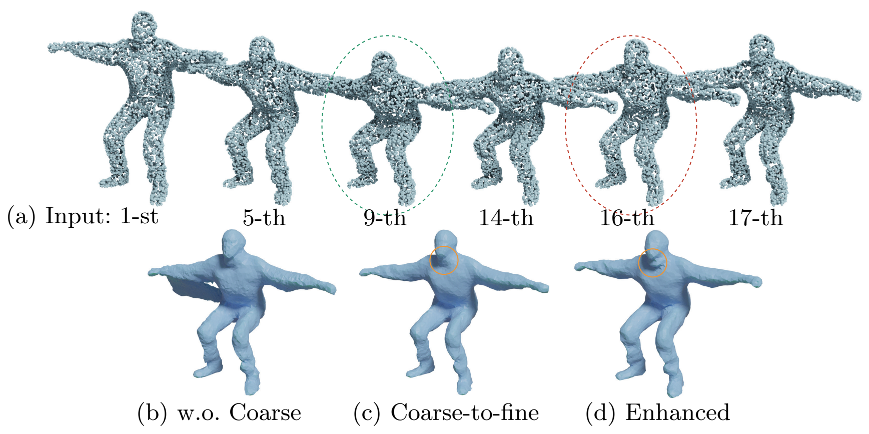


Fig. 1: Visual results of the template surface learned from different settings. For input point clouds, we use the green circle to mark the middle frame and the red circle to mark the template frame we selected.

⋆ Corresponding author. Email: `jh.hou@cityu.edu.hk`.

Arbitrarily long point cloud sequence. Our method is theoretically independent of the sequence length. We experimented with sequences of various lengths on "50007_shake_shoulders" sequence from DFAUST. As shown in Tab. 1 below, our method can effectively handle sequences of varying lengths. When the sequence is relatively long, the reconstruction quality decreases slightly due to the diversity of deformations.

Table 1: Results of our method on sequences with various lengths.

Sequence length	CD($\times 10^{-4}$) $\downarrow$	NC $\uparrow$	F-0.5% $\uparrow$	F-1% $\uparrow$	Corr. ($\times 10^{-2}$) $\downarrow$
10	0.154	0.957	0.866	0.992	0.53
30	0.138	0.954	0.876	0.994	0.97
50	0.167	0.955	0.831	0.989	1.46
80	0.265	0.946	0.704	0.960	1.72

Various data qualities. We experimented with point clouds of different qualities on the AMA dataset: (**1**) we randomly sampled N_k points from the ground truth mesh respectively as input of our method; and (**2**) we randomly sampled 5000 points from ground truth mesh, randomly selected 10 points from them, and deleted the h_k nearest neighbor points of these points to construct point clouds with holes. As listed in Tab. 2, our method shows robustness to variations in point cloud density and holes. We also showcase the visualization of constructed point clouds with various qualities in Fig. 2.

Table 2: Results of our method on point clouds with various qualities.

Input Data		CD($\times 10^{-4}$) $\downarrow$	NC $\uparrow$	F-0.5% $\uparrow$	F-1% $\uparrow$	Corr. ($\times 10^{-2}$) $\downarrow$
	$N_k = 2000$	0.451	0.906	0.563	0.899	3.54
(**1**)	$N_k = 3000$	0.402	0.911	0.610	0.916	3.46
	$N_k = 4000$	0.381	0.913	0.597	0.921	3.41
	$h_k = 10$	0.336	0.918	0.634	0.935	2.88
(**2**)	$h_k = 20$	0.366	0.914	0.617	0.926	3.57
	$h_k = 30$	0.405	0.911	0.587	0.912	3.50
GT-normal		0.318	0.918	0.656	0.940	3.10
Ours		0.324	0.919	0.649	0.939	2.89

Dependence on the normal direction. In all experiments, we did not use any ground-truth normals. Instead, we first use the algorithm in PyMeshLab [5] to

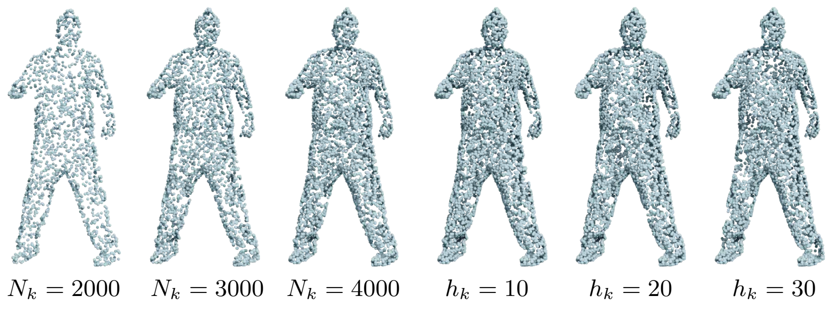


Fig. 2: Visualization of an input point cloud with various qualities.

estimate the normals of input point clouds. We then adjust the normal directions to ensure that the majority of normals point outward. Besides, we also replaced estimated normals with ground-truth normals in our method. As listed in Tab. 2, the performance is comparable to Ours, demonstrating the robustness of our method to normals of input point clouds.

Visualization of omitting loss terms. Additionally, Fig. 3 illustrates the visual results of our method when different loss terms are omitted. We can see that our method, incorporating all loss terms, achieves the best results.

2 More visual results

We present more visual results in this section and showcase the complete motion sequences in the *Video Demo*.

Comparisons with state-of-the-art methods. We present additional visual results for comparisons with LPDC [9] and Cadex [2] on the AMA dataset [10] (Fig. 4), DT4D dataset [3] (Fig. 5) and DFAUST dataset [1] (Fig. 6). It is evident that our method outperforms other approaches.

Performance on noisy data and partially missing data. Furthermore, in Fig. 7, we showcase more visual results of our method applied to noisy data constructed from the DFAUST dataset [1] and partially missing data constructed from AMA dataset [10], respectively.

3 Technical Details

In this section, we introduce the parameter settings and explain some definitions in the loss functions and evaluation metrics mentioned in the paper.

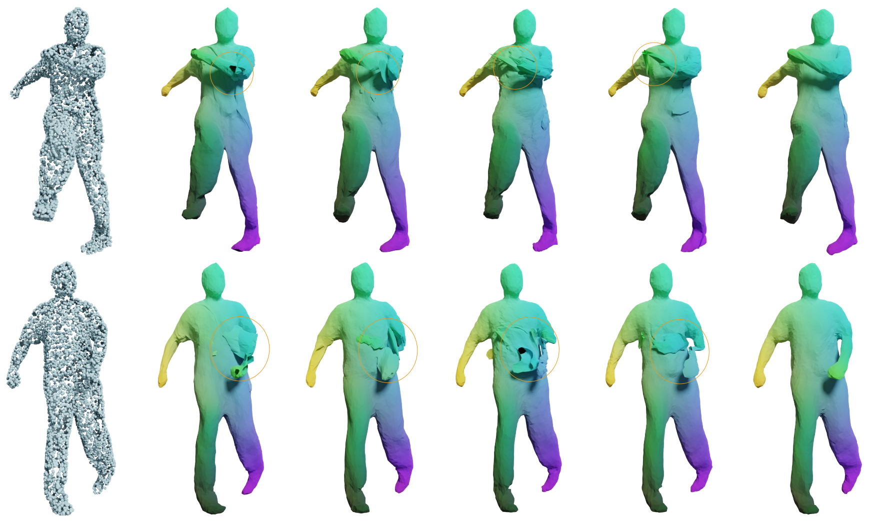
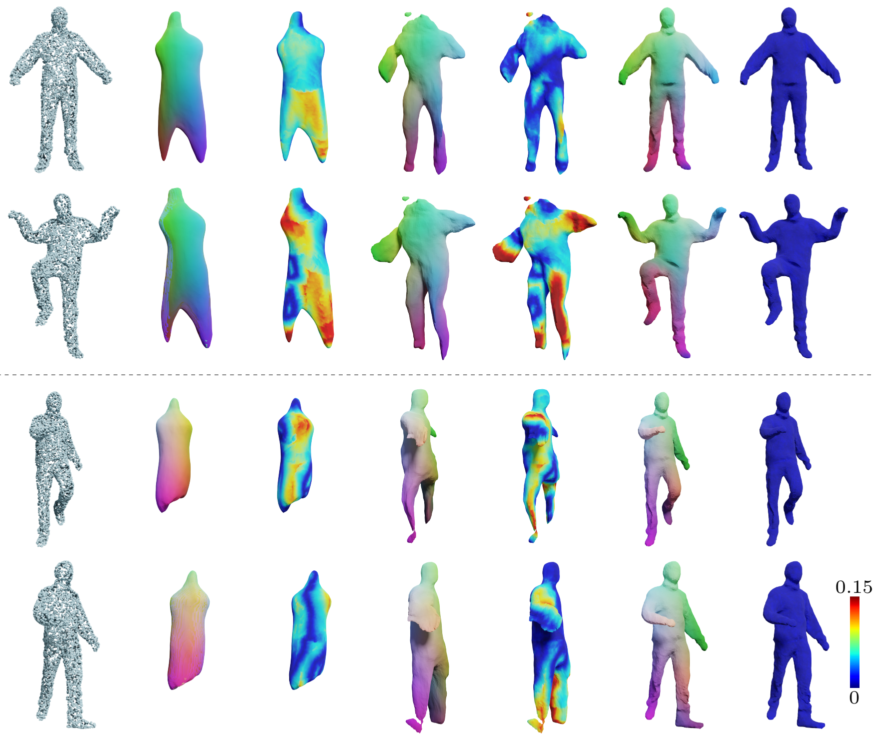


Fig. 3: Comparison of visual results by different variants of excluding a certain loss in our method. (a) Input; (b) w.o. $L_{\text{R-SDF}}$; (c) w.o. L_{Norm}; (d) w.o. L_{Smo}; (e) w.o. L_{Shape}; f) Ours.

Parameter settings. Reconstructing temporally consistent dynamic surfaces from point cloud sequences is challenging, especially for our unsupervised learning scenario. Therefore, we need multiple hyperparameters associated with the regularization terms for shrinking the large solution space of our unsupervised pipeline, we have uniformly set them to default values in all experiments and found that they generally work well for most examples, minimizing the need for additional parameter adjustments for new datasets. By default, in the calculation of the loss function (8), we set the number of the points for both $\widetilde{\mathbf{V}}_k$ in Eq. (9) and $\mathbf{Q}_s$ in Eq. (10) to be 10^4. We set $\eta = 0.1$ in Eq. (7), $\alpha = 5.56$ in Eq. (9) and $\beta = 50, \gamma = 10^2$ in Eq. (10). The default weights setting in the loss function are listed in Tab. 3. In addition, we also showcase the number of iterations and training time in this table.

Table 3: Default parameter settings and runtime. For the learning template surface, we listed iterations and running time for the coarse/fine stage.

Learning template surface					Temporal reconstruction						
$\tilde{w}_1$	$\tilde{w}_2$	$\tilde{w}_3$	#Iters	Time	w_1	w_2	w_3	w_4	w_5	#Iters	Time
5×10^2	10^{-3}	50	10^3 / 5×10^3	10 s/ 2 mins	5×10^2	10^{-3}	10^2	10^3	1	10^4	30 mins



(a) Input (b) LPDC [9] (c) Cadex [2] (d) Ours

Fig. 4: Comparison of visual results by different methods on the AMA dataset [10].

The details of loss functions. The Chamfer Distance between $\mathbf{X}$ and $\mathbf{Y}$ is defined as

$$\mathtt{CD}_{\ell_p}(\mathbf{X},\mathbf{Y}) = \frac{1}{|\widetilde{\mathbf{X}}|}\sum_{\mathbf{x}\in\widetilde{\mathbf{X}}} \|\mathbf{x}-\hat{\mathbf{y}}\|_{\ell_p} + \frac{1}{|\widetilde{\mathbf{Y}}|}\sum_{\mathbf{y}\in\widetilde{\mathbf{Y}}} \|\mathbf{y}-\hat{\mathbf{x}}\|_{\ell_p}\,, \tag{1}$$

where $p = 1, 2$. $\hat{\mathbf{y}} \in \widetilde{\mathbf{Y}}$ and $\hat{\mathbf{x}} \in \widetilde{\mathbf{X}}$ are the closest points for the $\mathbf{x}$ and $\mathbf{y}$ respectively. When $\mathbf{X}$(or $\mathbf{Y}$) is a point set, $\widetilde{\mathbf{X}} = \mathbf{X}$(or $\widetilde{\mathbf{Y}} = \mathbf{Y}$). When $\mathbf{X}$(or $\mathbf{Y}$) represents a mesh, $\widetilde{\mathbf{X}}$(or $\widetilde{\mathbf{Y}}$) denotes the sampling point on the mesh. In the learning phase, we set the number of sampling points to 10^4, and during the evaluation, we set the number of sampling points to 10^5.

In Eqs. (3) and (8), the normal consistency of $\mathbf{X}$ and $\mathbf{Y}$ is defined as

$$\mathtt{NC}_{\ell_1}(\mathbf{X},\mathbf{Y}) = \frac{1}{|\mathbf{N}_x|}\sum_{\mathbf{n}_x\in\mathbf{N}_x} |1-|\langle\mathbf{n}_x,\mathbf{n}_{\hat{y}}\rangle|| + \frac{1}{|\mathbf{N}_y|}\sum_{\mathbf{n}_y\in\mathbf{N}_y} |1-|\langle\mathbf{n}_y,\mathbf{n}_{\hat{x}}\rangle||\,, \tag{2}$$

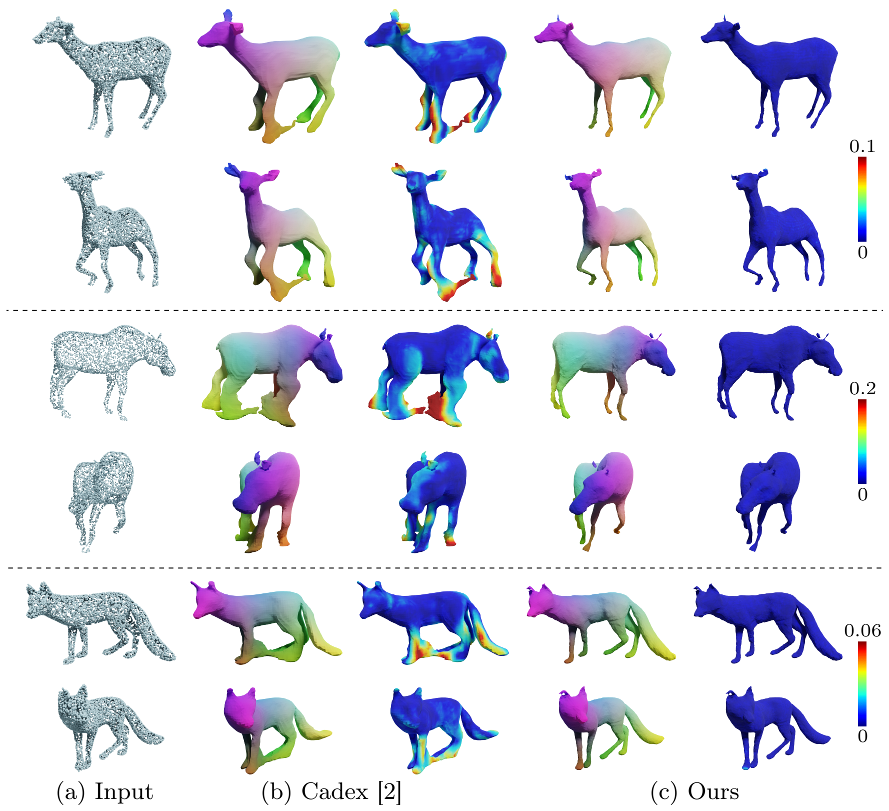


Fig. 5: Comparison of visual results by different methods on the DT4D dataset.

where $\mathbf{N}_x$ and $\mathbf{N}_y$ are the normal sets of $\widetilde{\mathbf{X}}$ and $\widetilde{\mathbf{Y}}$, respectively, $\mathbf{n}_{\hat{y}} \in \mathbf{N}_y$ and $\mathbf{n}_{\hat{x}} \in \mathbf{N}_x$ are the corresponding normals of $\hat{\mathbf{y}}$ and $\hat{\mathbf{x}}$ respectively. $\langle \cdot, \cdot \rangle$ is the inner product of two vectors.

In Eq. (4), the SDF value $\texttt{SDF}_{\text{IMLS}}(\mathbf{q}, \mathbf{P}_{k^*})$ at $\mathbf{q}$ approximated through implicit moving least-squares is defined as

$$\texttt{SDF}_{\text{IMLS}}(\mathbf{q}, \mathbf{P}_{k^*}) = \frac{\sum_{\mathbf{p}_j \in \mathcal{N}(\mathbf{q})} \theta(\|\mathbf{q} - \mathbf{p}_j\|, \zeta) \cdot \langle \mathbf{q} - \mathbf{p}_j, \mathbf{n}_j \rangle}{\sum_{\mathbf{p}_j \in \mathcal{N}(\mathbf{q})} \theta(\|\mathbf{q} - \mathbf{p}_j\|, \zeta)}, \tag{3}$$

where $\mathcal{N}(\mathbf{q})$ denotes the set of nearest points in $\mathbf{P}_{k^*}$. $\theta(d, \zeta) = \exp(-d^2/\zeta^2)$ is the weight of each point in $\mathcal{N}(\mathbf{q})$. Here we set $|\mathcal{N}(\mathbf{q})| = 10$ and $\zeta = 0.1$ by default.

Evaluation metrics. Following [4,7,8], the F-score to evaluate the reconstruction accuracy is defined as:

$$\text{F-score}(\mathcal{M}_k, \mathcal{M}_k^{\text{GT}}, \epsilon) = \frac{2 \cdot \text{Recall} \cdot \text{Precision}}{\text{Recall} + \text{Precision}}, \tag{4}$$

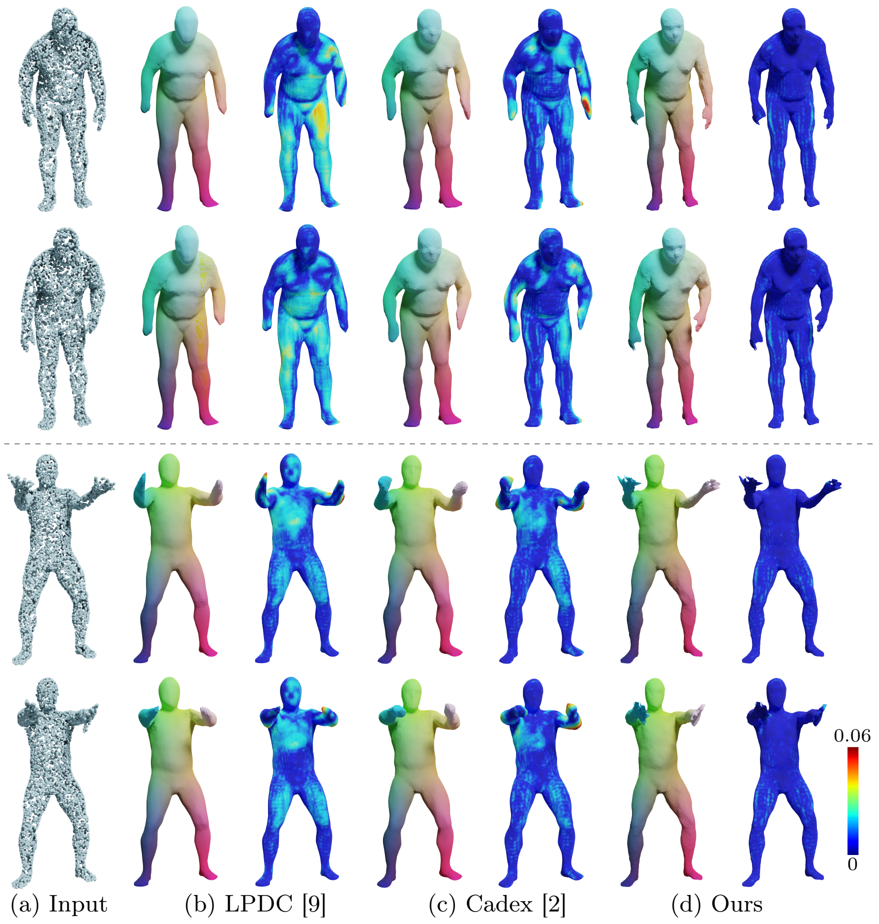


Fig. 6: Comparison of visual results by different methods on the DFAUST dataset.

where

$$\text{Recall}(\mathcal{M}_k, \mathcal{M}_k^{\text{GT}}, \epsilon) = \left| \left\{ \mathbf{p}_1 \in \widetilde{\mathbf{P}}_k, \text{s.t.} \min_{\mathbf{p}_2 \in \widetilde{\mathbf{P}}_k^{\text{GT}}} \|\mathbf{p}_1 - \mathbf{p}_2\| < \epsilon \right\} \right|,$$

$$\text{Precision}(\mathcal{M}_k, \mathcal{M}_k^{\text{GT}}, \epsilon) = \left| \left\{ \mathbf{p}_2 \in \widetilde{\mathbf{P}}_k^{\text{GT}}, \text{s.t.} \min_{\mathbf{p}_1 \in \widetilde{\mathbf{P}}_k} \|\mathbf{p}_1 - \mathbf{p}_2\| < \epsilon \right\} \right|.$$

Here $\mathcal{M}_k^{\text{GT}}$ is the k-th ground-truth mesh, $\widetilde{\mathbf{P}}_k(\widetilde{\mathbf{P}}_k^{\text{GT}})$ is the point set of the randomly sampling 10^5 points from $\mathcal{M}_k(\mathcal{M}_k^{\text{GT}})$. We set ϵ to 0.5% or 1%, and compute mean value of F-score$(\mathcal{M}_k, \mathcal{M}_k^{\text{GT}}, \epsilon)$ $(k = 1, 2, ..., K)$ to get F-0.5% or F-1%.

We denote $\{\mathbf{P}^{\mathrm{GT}}(k)\}_k$ as the temporally consistent corresponding points on the ground-truth surfaces. We set $|\mathbf{P}^{\mathrm{GT}}(k)| = 10^5$. In the first frame, we construct the index set of nearest points:

$$\mathtt{Index} = \{\rho_i | \mathbf{v}_{\rho_i} = \underset{\mathbf{v} \in \mathbf{V}_1}{\arg\min} \|\mathbf{v} - \mathbf{p}_i^{\mathrm{GT}}(1)\|\},$$

where $\mathbf{V}_1$ is the vertex set of 1-st reconstructed mesh $\mathcal{M}_1$, $\mathbf{p}_i^{\mathrm{GT}}(1) \in \mathbf{P}^{\mathrm{GT}}(1)$. Following [2, 6], the Correspondences Error is defined as

$$\text{Corr.} = \frac{1}{|\mathtt{Index}| \cdot K} \sum_{k=1}^{K} \sum_{\rho_i \in \mathtt{Index}} \|\mathbf{v}_{\rho_i}^k - \mathbf{p}_i^{\mathrm{GT}}(k)\|.$$

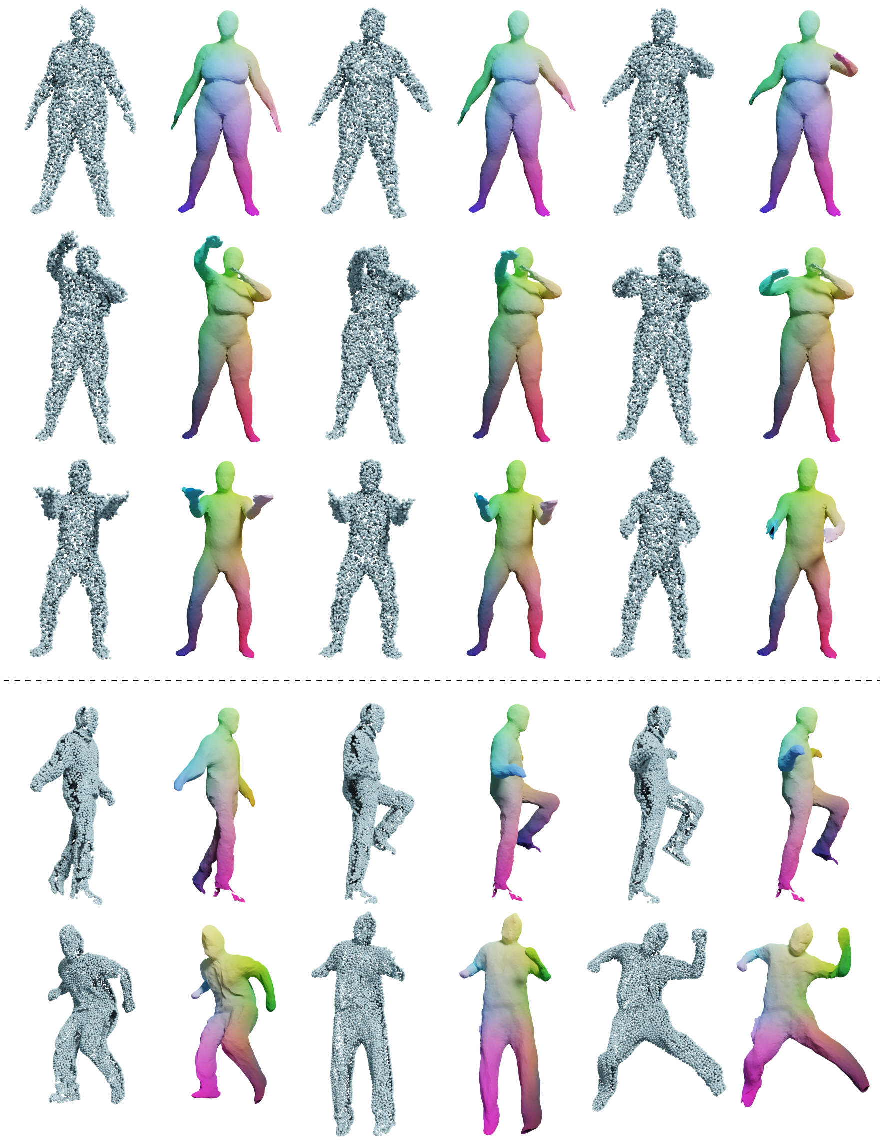


Fig. 7: Visual results of our method on constructed noisy data from the DFAUST dataset [1] (Top) and partially missing data from AMA dataset [10] (Bottom).

References

1. Bogo, F., Romero, J., Pons-Moll, G., Black, M.J.: Dynamic faust: Registering human bodies in motion. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 6233–6242 (2017)
2. Lei, J., Daniilidis, K.: Cadex: Learning canonical deformation coordinate space for dynamic surface representation via neural homeomorphism. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 6624–6634 (2022)
3. Li, Y., Takehara, H., Taketomi, T., Zheng, B., Nießner, M.: 4dcomplete: Non-rigid motion estimation beyond the observable surface. Int. Conf. Comput. Vis. (2021)
4. Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A.: Occupancy networks: Learning 3d reconstruction in function space. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 4460–4470 (2019)
5. Muntoni, A., Cignoni, P.: PyMeshLab (Jan 2021). `https://doi.org/10.5281/zenodo.4438750`
6. Niemeyer, M., Mescheder, L., Oechsle, M., Geiger, A.: Occupancy flow: 4d reconstruction by learning particle dynamics. In: Int. Conf. Comput. Vis. pp. 5379–5389 (2019)
7. Peng, S., Jiang, C., Liao, Y., Niemeyer, M., Pollefeys, M., Geiger, A.: Shape as points: A differentiable poisson solver. Adv. Neural Inform. Process. Syst. **34**, 13032–13044 (2021)
8. Ren, S., Hou, J., Chen, X., He, Y., Wang, W.: Geoudf: Surface reconstruction from 3d point clouds via geometry-guided distance representation. In: Int. Conf. Comput. Vis. pp. 14214–14224 (2023)
9. Tang, J., Xu, D., Jia, K., Zhang, L.: Learning parallel dense correspondence from spatio-temporal descriptors for efficient and robust 4d reconstruction. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 6022–6031 (2021)
10. Vlasic, D., Baran, I., Matusik, W., Popović, J.: Articulated mesh animation from multi-view silhouettes. ACM Trans. Graph. **27**(3), 97:1–97:9 (Aug 2008)