Eurographics Symposium on Geometry Processing 2023
P. Memari and J. Solomon
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 42 (2023), Number 5

Deep Deformation Detail Synthesis for Thin Shell Models –*Supplementary Material*–

Lan Chen[1,2], Lin Gao[†1,3], Jie Yang[1,3], Shibiao Xu[†4], Juntao Ye[2], Xiaopeng Zhang[2] and Yu-Kun Lai[5]

[1]University of Chinese Academy of Science, China
[2]Institute of Automation, Chinese Academy of Sciences, China
[3]Institute of Computing Technology, Chinese Academy of Sciences, China
[4]Beijing University of Posts and Telecommunications, China
[5]Cardiff University, United Kingdom

1. Overview

This supplementary material accompanies the main paper, which presents more technical and network architecture details, dataset implementation, and more results of perception study and ablation study.

All sections are listed as follows:

- Section 2 provides more details of deformation representation, DeformTransformer networks and the collision refinement method;
- Section 3 provides the datasets preparation and more training/implementation details for our network;
- Section 4 provides more results of qualitative evaluation, perception study, and ablation study including spatial temporal module evaluation and evaluation of coarse meshes.

2. Approach

2.1. Deformation Representation

Take coarse meshes $\mathcal{C}$ for instance and fine meshes $\mathcal{D}$ are processed in the same way. Assume that a sequence of coarse meshes contains n models with the same connectivity, each denoted as $\mathcal{C}_t$ $(1 \le t \le n)$. A mesh with the same topology is chosen as the reference model, denoted as $\mathcal{C}_0$. For example, for garment animation, this can be the garment mesh worn by a character in the T pose. $\mathbf{p}_{t,i} \in \mathbb{R}^3$ is the i^{th} vertex on the t^{th} mesh. To represent the local shape deformation, the deformation gradient $\mathbf{T}_{t,i} \in \mathbb{R}^{3\times 3}$ can be obtained by minimizing the following energy:

$$\arg\min_{\mathbf{T}_{t,i}} \sum_{j\in\mathcal{N}_i} c_{ij} \|(\mathbf{p}_{t,i}-\mathbf{p}_{t,j}) - \mathbf{T}_{t,i}(\mathbf{p}_{0,i}-\mathbf{p}_{0,j})\|_2^2 \tag{1}$$

where $\mathcal{N}_i$ is the one-ring neighbors of the i^{th} vertex, and c_{ij} is the cotangent weight $c_{ij} = \cot\alpha_{ij} + \cot\beta_{ij}$ [SHA07, LG14], where α_{ij} and β_{ij} are angles opposite to the edge connecting the i^{th} and j^{th} vertices.

The main drawback of the deformation gradient representation is that it cannot handle large-scale rotations, which often happen in cloth animation. Using polar decomposition, the deformation gradient $\mathbf{T}_{t,i}$ can be decomposed into a rotation part and a scaling/shearing part $\mathbf{T}_{t,i} = \mathbf{R}_{t,i}\mathbf{S}_{t,i}$. The scaling/shearing transformation $\mathbf{S}_{t,i}$ is uniquely defined, while the rotation $\mathbf{R}_{t,i}$ corresponds to infinite possible rotation angles (differed by multiples of 2π, along with possible opposite orientation of the rotation axis). Typical formulation often constrain the rotation angle to be within $[0,\pi]$ which is unsuitable for smooth large-scale animations.

In order to handle large-scale rotations, we first require the orientations of rotation axes and rotation angles of spatially adjacent vertices on the same mesh to be as consistent as possible. Especially for our sequence data, we further add constraints for adjacent frames to ensure the temporal consistency of the orientations of rotation axes and rotation angles on each vertex. We first consider consistent orientation for axes.

$$\begin{aligned}\arg\max_{o_{t,i}} & \sum_{(i,j)\in\mathcal{E}} o_{t,i}o_{t,j}\cdot s(\omega_{t,i}\cdot\omega_{t,j},\theta_{t,i},\theta_{t,j}) \\ & + \sum_{i\in\mathcal{V}} o_{t,i}\cdot s(\omega_{t,i}\cdot\omega_{t-1,i},\theta_{t,i},\theta_{t-1,i}) \\ & \text{s.t.} \quad o_{t,1}=1, o_{t,i}=\pm 1 (i\neq 1)\end{aligned} \tag{2}$$

where t is the index of the frame, $\mathcal{E}$ is the edge set, and $\mathcal{V}$ is the vertex set. Denote by $(\omega_{t,i},\theta_{t,i})$ one possible choice for the rotation axis and rotation angle that match $\mathbf{R}_{t,i}$. $o_{t,i} \in \{+1,-1\}$ specifies whether the rotation axis is flipped ($o_{t,i}=1$ if the rotation axis is unchanged, and -1 if its opposite is used instead). The first term promotes spatial consistency while the second term promotes temporal consistency.

s is a function measuring orientation consistency, which is defined as follows:

$$s(\omega_{t,i},\omega_{t,j}) = \begin{cases} 0, & |\omega_{t,i}\cdot\omega_{t,j}| \le \varepsilon_1 \text{ or } \theta_{t,i}<\varepsilon_2 \text{ or } \theta_{t,j}<\varepsilon_2 \\ 1, & \text{Otherwise if } \omega_{t,i}\cdot\omega_{t,j} > \varepsilon_1 \\ -1, & \text{Otherwise if } \omega_{t,i}\cdot\omega_{t,j} < -\varepsilon_1 \end{cases} \tag{3}$$

The first case here is to ignore settings where the rotation angle is

near zero, as the rotation axis is not well defined in such cases. As for rotation angles, we optimize the following

$$\begin{aligned}\arg\min_{r_{t,i}} & \sum_{(i,j)\in\mathcal{E}} \|(r_{t,i}\cdot 2\pi + o_{t,i}\theta_{t,i}) - (r_{t,j}\cdot 2\pi + o_{t,j}\theta_{t,j})\|_2^2 \\ & + \sum_{i\in\mathcal{V}} \|(r_{t,i}\cdot 2\pi + o_{t,i}\theta_{t,i}) - (r_{t-1,i}\cdot 2\pi + o_{t,j}\theta_{t-1,i})\|_2^2 \\ & \text{s.t.} \quad r_{t,i}\in\mathbb{Z}, \;\; r_{t,1}=0. \end{aligned} \tag{4}$$

where $r_{t,i}\in\mathbb{Z}$ specifies how many 2π rotations should be added to the rotation angle. The two terms here promote spatial and temporal consistencies of rotation angles, respectively. These optimizations can be solved using integer programming, and we use the mixed integer CoMISo [BZK09] which provides an efficient solver. See [GLY*19] for more details. A similar process is used to compute the TS-ACAP representation of the fine meshes.

2.2. DeformTransformer Networks

Unlike [TGLX18, WSFM19] which use fully connected layers for mesh encoder, we perform convolutions on meshes to learn to extract useful features using compact shared convolutional kernels. We use a convolution operator on vertices [DMI*15, TGL*22] where the output at a vertex is obtained as a linear combination of input in its one-ring neighbors along with a bias. The input to our network is the TS-ACAP representation, which for the i^{th} vertex of the t^{th} mesh, we collect non-trivial coefficients from the rotation $\mathbf{R}_{t,i}$ and scaling/shearing $\mathbf{S}_{t,i}$, which forms a 9-dimensional feature vector (see [GLY*19] for more details). Denote by $\mathbf{f}_i^{(k-1)}$ and $\mathbf{f}_i^k$ the feature of the i^{th} vertex at the $(k-1)^{\text{th}}$ and k^{th} layers, respectively. The convolution operator is defined as follows:

$$\mathbf{f}_i^{(k)} = \mathbf{W}_{point}^{(k)}\cdot\mathbf{f}_i^{(k-1)} + \mathbf{W}_{neighbor}^{(k)}\cdot\frac{1}{D_i}\sum_{j=1}^{D_i}\mathbf{f}_{n_{ij}}^{(k-1)} + \mathbf{b}^{(k)} \tag{5}$$

where $\mathbf{W}_{point}^{(k)}$, $\mathbf{W}_{neighbor}^{(k)}$ and $\mathbf{b}^{(k)}$ are learnable parameters for the k^{th} convolutional layer, D_i is the degree of the i^{th} vertex, $n_{ij}(1\le j\le D_i)$ is the j^{th} neighbor of the i^{th} vertex. Each convolutional layer is followed by a layer normalization and leaky ReLU activation function.

2.3. Collision Refinement

Our training data is generated by PBS and is collision-free. Since human body (or other obstacles) information is unseen in our algorithm, it does not guarantee the predicted cloth is free from any penetration. Especially for tight garment like T-shirts, it will be apparent if collision between the garment and human body happens. We use a fast refinement method [WSFM19] to push the cloth vertices colliding with the body outside while preserving the local wrinkle details (see Fig. 1). For each vertex detected inside the body, we find its closest point over the body surface with normal and position. Then the cloth mesh is deformed to update the vertices by minimizing the energy which penalizes the euclidean distance and Laplacian difference between the updated mesh and the initial one (please refer to [WSFM19] for details). The collision solving process usually takes less than 3 iterations to converge to a collision-free state.

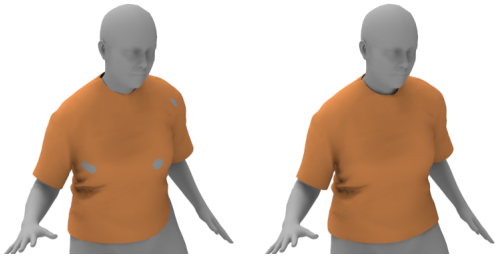


Figure 1: For tight clothing, data-driven cloth deformations may suffer from apparent collisions with the body (left). We apply a simple postprocessing step to push the collided T-shirt vertices outside the body (right).

3. Implementation

We describe the details of the dataset construction and the network architecture in this section.

Datasets. To test our method, we construct 5 datasets, called TSHIRT, PANTS, SKIRT, SHEET and DISK respectively (shown in Fig. 2). The former three datasets are different types of garments, *i.e.*, T-shirts, skirts and pants worn on human bodies. Each type of garment is represented by both low-resolution and high-resolution meshes, containing 246 and 14,190 vertices for the T-shirts, 219 and 12,336 vertices for the skirts, 200 and 11,967 vertices for the pants. Garments of the same type and resolution are simulated from a template mesh, which means such meshes obtained through cloth animations have the same number of vertices and the same connectivity. These garments are dressed on animated characters, which are obtained via driving a body in the SMPL (Skinned Multi-Person Linear) model [LMR*15] with publicly available motion capture data from CMU [Hod15]. Since the motion data is captured, there are some self-collisions or long repeated sequences. After removing poor quality data, we select various motions, such as dancing, walking, running, jumping, etc., including 20 sequences (9031, 6134, 7680 frames in total for TSHIRT, PANTS and SKIRT respectively). In these motions, 18 sequences are randomly selected for training and the remaining 2 sequences for testing. The SHEET dataset consists of a pole or a sphere of three different sizes crashing to a piece of cloth sheet. The coarse mesh has 81 vertices and the fine mesh has 4,225 vertices. There are 4,000 frames in the SHEET dataset, in which 3200 frames for training and the remaining 800 frames for testing. We construct the DISK dataset by draping a round tablecloth to a cylinder in the wind, with 148 and 7,729 vertices for coarse and fine meshes respectively. We adjust the velocity of the wind to get various animation sequences, in which 1600 frames for training and 400 frames for testing.

It is difficult to gather real-world data, we note that most data-driven methods construct their high-resolution databases using physics-based methods. We also use PBS to prepare the above datasets, both low-resolution (LR) and high-resolution (HR) cloth animations. The initial state of the HR mesh is obtained by applying the Loop subdivision scheme [Loo87] to the coarse mesh and

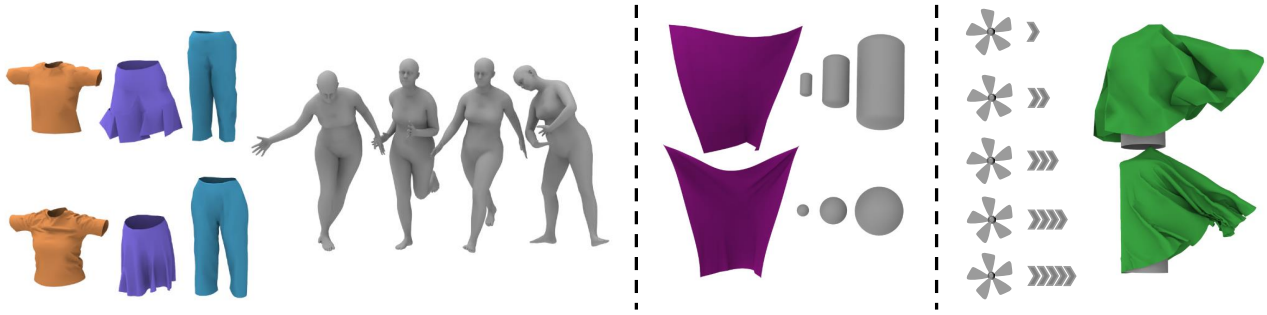


Figure 2: We test our algorithm on 5 datasets including TSHIRT, PANTS, SKIRT, SHEET and DISK. The former three are garments (T-shirts, skirts, and pants) dressed on a template body and simulated with various motion sequences. The SHEET dataset is a square sheet interacting with various obstacles. The DISK dataset is a round tablecloth draping on a cylinder in the wind of various velocities. Each cloth shape has a coarse resolution (top) and a fine resolution (bottom).

waiting for several seconds till stable. Previous works [KGBS11, ZBO13, CYJ*18] usually constrain the high-resolution meshes by various tracking mechanisms to ensure that the coarse cloth can be seen as a low-resolution version of the fine cloth during the complete animation sequences. However, fine-scale wrinkle dynamics cannot be captured by this model, as wrinkles are defined quasistatically and limited to a constrained subspace. Thus we instead perform PBS for the two resolution meshes *separately*, without any constraints between them. We use an open-source cloth simulation engine called ARCSim [NSO12] to produce all animation sequences of low- and high-resolution meshes with the same parameter setting. In our experiment, we choose the Gray Interlock from a library of measured cloth materials [WOR11] as the material parameters for ARCSim simulation. Specially for garments interacting with characters, to ensure collision-free, we manually put the coarse and fine garments on a template human body (in the T pose) and run the simulation to let the clothing relax.

The initial poses in the CMU dataset for all subsequent simulations may be different. Thus to ensure collision-free state we interpolate 15 frames between the T pose and the initial pose of each motion sequence. Before applying the motion sequence for simulation, the sequence is smoothed using a convolution operation.

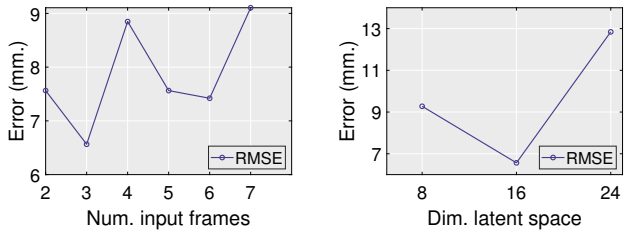


Figure 3: Evaluation of hyperparameters in the Transformer network, using the SKIRT dataset. (Left) average error for the reconstructed results as a function of the number of input frames. (Right) error for the synthesized results under the condition of various dimensions of the latent layer.

Network architecture. Our transduction network consists of two components, namely convolutional encoders to map coarse and fine mesh sequences into latent spaces for improved generalization capability, and the Transformer network for spatio-temporally coherent deformation transduction. We add random normal noise of zero means and fixed variance (0.001) to the coarse inputs in the training phase for data augmentation. The feature encoder module takes the 9-dimensional TS-ACAP features defined on vertices as input, followed by two convolutional layers with *tanh* as the activation function. In the last convolutional layer we abandon the activation function, similar to [TGL*22]. A fully connected layer is used to map the output of the convolutional layers into a 16-dimensional latent space. We train one encoder for coarse meshes and another for fine meshes separately. For the DeformTransformer network, its input includes the embedded latent vectors from both the coarse and fine domains. The DeformTransformer network consists of spatial-temporal encoders and decoders, each including a stack of 2 identical blocks with 8-head attention. Different from variable length sequences used in natural language processing, we fix the number of input frames (to 3 in our experiments) since a motion sequence may include a thousand frames and calculating attention to all previous frames causes memory explosion. We perform experiments to evaluate the performance of our method with different settings. As shown in Fig. 3 (left), using 3 input frames is found to perform well in our experiments. We also evaluate the results generated with various dimensions of latent space shown in Fig. 3 (right). When the dimension of latent space is larger than 16, the network can easily overfit. Thus we set the dimension of the latent space to 16, which is sufficient for all the examples in the paper. We use Adam optimization and batch normalization. The learning rate is set to 0.001 initially and times 0.1 every 20 epochs. The overall epoch is set to 200. Table 1 shows the average per-frame execution time of our method for various cloth datasets, including coarse simulation, TS-ACAP extraction, synthesis of high-resolution TS-ACAP and coordinates, and collision refinement.

Table 1: Statistics and timing (sec/frame) of the testing examples including five types of thin shell animations.

Benchmark	#verts LR	#verts HR	ARCSim HR	ours	speedup	our components			
						coarse sim.	TS-ACAP extraction	synthesizing (GPU)	refinement
TSHIRT	246	14,190	8.72	0.867	**10**	0.73	0.11	0.012	0.015
PANTS	200	11,967	10.92	0.904	**12**	0.80	0.078	0.013	0.013
SKIRT	127	6,812	6.84	0.207	**33**	0.081	0.10	0.014	0.012
SHEET	81	4,225	2.48	0.157	**16**	0.035	0.10	0.011	0.011
DISK	148	7,729	4.93	0.139	**35**	0.078	0.041	0.012	0.008

Table 2: User study results on cloth detail synthesis. We show the average ranking score of the five methods: Chen *et al.* [CZY21] w/ track, Chen *et al.* w/o track, Zurdo *et al.* [ZBO13] w/ track, Zurdo *et al.* w/o track, and ours. The ranking ranges from 1 (the best) to 5 (the worst). The results are calculated based on 370 trials. We see that our method achieves the best in terms of wrinkles, temporal stability and overall quality.

Method	Wrinkles	Temporal stability	Overall
Chen *et al.* w/ track	2.184	2.4365	2.1453
Chen *et al.* w/o track	4.653	3.4443	3.436
Zurdo *et al.* w/ track	2.3434	4.3240	3.458
Zurdo *et al.* w/o track	4.3096	4.5215	4.986
Ours	**1.3482**	**1.2763**	**1.3012**

4. Results

4.1. Qualitative Evaluation

We show more results of loose garments and free-flying cloth, with comparisons on the SKIRT, SHEET, and DISK datasets (shown in Fig. 4.

4.2. Perception Study

We further conduct a user study to evaluate the stability and realism of the synthesized dense mesh dynamics. 37 volunteers are involved in this user study. For every question, we give one sequence and 5 images of coarse meshes as references, and then let the user rank the corresponding outputs from Chen *et al.* [CZY21] w/ and w/o track, Zurdo *et al.* [ZBO13] w/ and w/o track and ours according to three different criteria (wrinkles, temporal stability and overall). We shuffle the order of the algorithms each time we exhibit the question and show shapes from the five methods randomly to avoid bias. We show the results of the user study in Table 2, where we observe that our generated shapes perform the best on all three criteria.

Table 3: Average RMSE values of our method for three scale factors of $\times 16$, $\times 64$, $\times 256$ of face number on the SHEET datasets.

	$\times 16$	$\times 64$	$\times 256$
SHEET	0.00589	0.00543	0.00531

4.3. Ablation Study

Spatial Temporal Module Evaluation. Since the key components of our network are the spatial and temporal modules, we evaluate the impact of each module and show more corresponding results in Fig. 5.

Evaluation of Coarse Meshes. To study the impact of the resolution of different input coarse meshes, we conduct experiments on the SHEET dataset. There are three different input meshes with 512, 128, 32 faces, corresponding to the face scale factors $\times 16$, $\times 64$, $\times 256$, respectively. With these inputs and the high resolution meshes of 8192 faces, we train three models separately. As shown in Fig. 6, the synthesized results even for high scale factor $\times 256$ show that our method can predict detailed meshes with appropriate wrinkles. While the less the scale factor is, the higher similarity between synthesized meshes and the ground truth is. For quantitatively evaluation, the RMSE values of different scale factors are shown in Table 3.

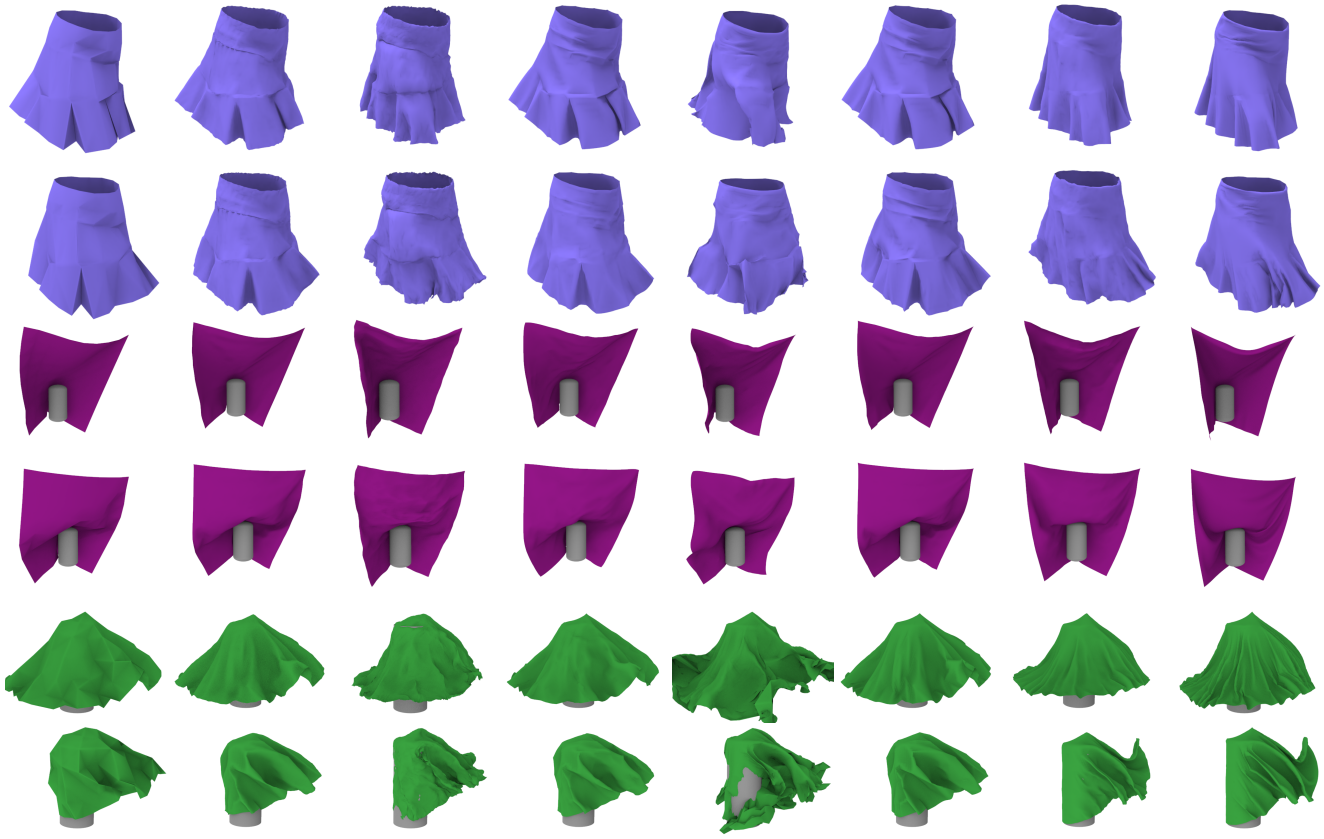
References

[BZK09] BOMMES D., ZIMMER H., KOBBELT L.: Mixed-integer quadrangulation. *ACM Trans. Graph. 28*, 3 (July 2009). 2

[CYJ*18] CHEN L., YE J., JIANG L., MA C., CHENG Z., ZHANG X.: Synthesizing cloth wrinkles by CNN-based geometry image superresolution. *Computer Animation and Virtual Worlds 29*, 3-4 (2018), e1810. 3

[CZY21] CHEN L., ZHANG X., YE J.: Multi-feature super-resolution network for cloth wrinkle synthesis. *J. Comput. Sci. Technol. 36* (2021), 478–493. 4, 5

[DMI*15] DUVENAUD D. K., MACLAURIN D., IPARRAGUIRRE J., BOMBARELL R., HIRZEL T., ASPURU-GUZIK A., ADAMS R. P.: Convolutional networks on graphs for learning molecular fingerprints. In *Advances in neural information processing systems* (2015), pp. 2224–2232. 2



(a) Input | (b) Chen *et al.* w/ track | (c) Chen *et al.* w/o track | (d) Zurdo *et al.* w/ track | (e) Zurdo *et al.* w/o track | (f) Tracked | (g) Ours | (h) GT

Figure 4: Comparison of the reconstruction results for unseen data in the datasets of loose garments. (a) the coarse simulation, (b) the results of Chen *et al.* [CZY21], (c) the results of Zurdo *et al.* [ZBO13], (d) the results generated by physics-based tracking simulation [CZY21], (e) our results, (f) the ground truth generated by PBS.

[GLY*19] GAO L., LAI Y.-K., YANG J., LING-XIAO Z., XIA S., KOBBELT L.: Sparse data driven mesh deformation. *IEEE Transactions on Visualization and Computer Graphics* (2019). 2

[Hod15] HODGINS J.: CMU graphics lab motion capture database, 2015. 2

[KGBS11] KAVAN L., GERSZEWSKI D., BARGTEIL A. W., SLOAN P.-P.: Physics-inspired upsampling for cloth simulation in games. *ACM Trans. Graph. 30*, 4 (2011), 93:1–93:10. 3

[LG14] LEVI Z., GOTSMAN C.: Smooth rotation enhanced as-rigid-as-possible mesh animation. *IEEE Transactions on Visualization and Computer Graphics 21*, 2 (2014), 264–277. 1

[LMR*15] LOPER M., MAHMOOD N., ROMERO J., PONS-MOLL G., BLACK M. J.: SMPL: A skinned multi-person linear model. *ACM transactions on graphics (TOG) 34*, 6 (2015), 1–16. 2

[Loo87] LOOP C.: *Smooth Subdivision Surfaces Based on Triangles*. Master's thesis, University of Utah, 1987. 2

[NSO12] NARAIN R., SAMII A., O'BRIEN J. F.: Adaptive anisotropic remeshing for cloth simulation. *ACM Trans. on Graph. 31*, 6 (2012), 147:1–10. 3

[SHA07] SORKINE-HORNUNG O., ALEXA M.: As-rigid-as-possible surface modeling. In *Symposium on Geometry Processing* (2007), vol. 4, pp. 109–116. 1

[TGL*22] TAN Q., GAO L., LAI Y., YANG J., XIA S.: Mesh-based autoencoders for localized deformation component analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence 44*, 10 (2022), 6297–6310. 2, 3

[TGLX18] TAN Q., GAO L., LAI Y.-K., XIA S.: Variational autoencoders for deforming 3D mesh models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2018). 2

[WOR11] WANG H., O'BRIEN J. F., RAMAMOORTHI R.: Data-driven elastic models for cloth: modeling and measurement. *ACM Trans. Graph. 30*, 4 (2011), 71:1–71:12. 3

[WSFM19] WANG T. Y., SHAO T., FU K., MITRA N. J.: Learning an intrinsic garment space for interactive authoring of garment animation. *ACM Transactions on Graphics (TOG) 38*, 6 (2019), 220. 2

[ZBO13] ZURDO J. S., BRITO J. P., OTADUY M. A.: Animating wrinkles by example on non-skinned cloth. *IEEE Trans. Visual. Comput. Graph. 19*, 1 (2013), 149–158. 3, 4, 5

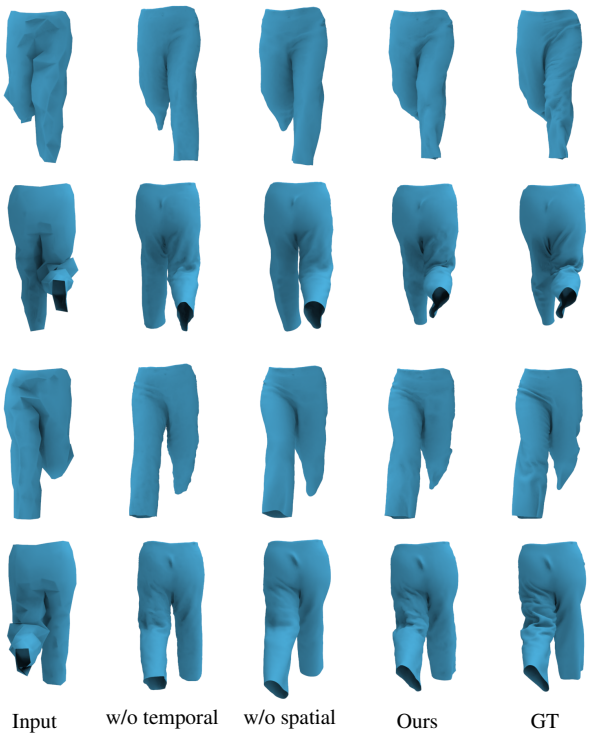


Figure 5: Ablation study of network architecture."w/o spatial" and "w/o temporal" are our method without the spatial inference module, and without temporal coherence module, respectively.

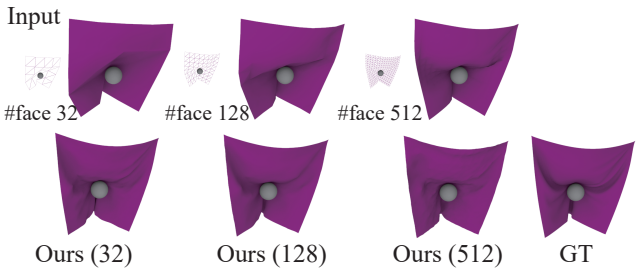


Figure 6: Comparison of the synthesized meshes in the SHEET dataset feeding various coarse inputs of different resolutions.