Lan Chen, Juntao Ye, Xiaopeng Zhang. Journal of computer science and technology: Instruction for authors. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 33(1): 1–4 January 2021. DOI 10.1007/s11390-021-1331-y

Multi-Feature Super-Resolution Network for Cloth Wrinkle Synthesis Supplementary Material

The supplementary material of the paper "Multi-Feature Super-Resolution Network for Cloth Wrinkle Synthesis" consists of the following components:

(1) This document provides the details about feature descriptors in geometry images, rigid motion transformation and the kinematics-based loss function.

(2) The supplemental video shows a range of different examples of synthesized results, as well as comparisons with [13], including animated sequences.

1 Feature descriptors

The state of a cloth mesh $\mathbf{M}$, when embedded in 3D space, can be denoted as the positions of vertices $\mathbf{x} = [\mathbf{x}(q_1), \mathbf{x}(q_2), \cdots, \mathbf{x}(q_m)] \in \mathbb{R}^{3m}$. A physical simulation usually starts from a 3D space embedding $\mathbf{x}_0$ of the rest state mesh $\mathbf{M}_0$. Then we generate a sequence of cloth meshes, denoted as $\mathbf{x}_0, \mathbf{x}_1, \cdots, \mathbf{x}_N$. We design a feature descriptor by considering the *position*, the *normal* and the *velocity* of a sampling point.

The displacement. The geometry images technique [18] converts world space coordinates into three grayscale values (as RGB values). The work of [13] followed this paradigm. However as we are only interested in the intrinsic shape of the mesh but not its spatial locations, directly encoding the absolute coordinates is not recommended. We make an adaption by encoding displacement vectors instead.

The displacement vector $\mathbf{d}_k(q_i)$ is defined, for the i-th vertex, as the difference between its position at the frame k and its starting position: $\mathbf{d}_k(q_i) = \mathbf{x}_k(q_i) - \mathbf{x}_0(q_i)$. This vector is then converted into another format such as RGB values. Displacement for any sample point inside a triangle face is interpolated from mesh vertices.

The normal. Wrinkles are closely related to dihedral angles, which are defined as the angle between two normal vectors of adjacent faces. In this sense, normals are also very important features in describing a shape. We compute the normal for vertex q_i as area-weighted average of normals of the faces adjacent to q_i:

$$\mathbf{n}(q_i) = \frac{\sum_{T\in\mathcal{N}(q_i)} \alpha_T \mathbf{n}(T)}{\left\|\sum_{T\in\mathcal{N}(q_i)} \alpha_T \mathbf{n}(T)\right\|}, \tag{1}$$

where $\mathcal{N}(q_i)$ denotes the set of faces incident to vertex q_i, T is a triangle belonging to $\mathcal{N}(q_i)$, α_T is the area of triangle T and $\mathbf{n}(T) \in \mathbb{R}^3$ is the unit vector for triangle normal.

The velocity. In an animation, the wrinkling behavior exhibits smooth transition from frame to frame. Temporal coherence needs to be considered, otherwise cloth motion will show obvious jittering effects. Temporally coherent features should not vary dramatically from frame to frame, so extra constraints are needed to achieve this goal. Fortunately, for physically simulated cloth frame data, there is a natural connection between two consecutive frames: the vertex velocities. Velocities at the frame k predict the positions at the frame $k+1$, which are computed as:

$$\mathbf{v}_k = (\mathbf{x}_{k+1} - \mathbf{x}_k)/\Delta t. \tag{2}$$

Thus velocity is added to the feature vector. As above the velocity of arbitrary sample point is interpolated from the velocity of vertices.

2 Rigid motion invariant features

When a mesh changes its state from $\mathbf{x}_0$ to $\mathbf{x}_k$, the position change can be decomposed into a rigid motion and an intra-mesh non-rigid deformation. The rigid motion, represented by an optimal rotation $\mathbf{R}$ and translation $\mathbf{t}$, has no contribution to the wrinkling behavior thus is to be culled off. The Kabsch Algorithm [47] is used to find the $\mathbf{R}$ and $\mathbf{t}$ which minimize the mean-square-error between two point sets at the frame k and its starting state:

$$\text{argmin} \sum_{i=1}^{m} \parallel (\mathbf{R}\mathbf{x}_k(q_i) + \mathbf{t}) - \mathbf{x}_0(q_i) \parallel^2 \text{ s.t. } \det(\mathbf{R}) = 1. \quad (3)$$

To reduce computation cost, we only compute rigid motion of LR meshes at each frame and apply the same $(\mathbf{R}, \mathbf{t})$ to the corresponding HR meshes. With the computed $\mathbf{R}$ and $\mathbf{t}$, the rigid motion invariant features in the paper are calculated of the form

$$\mathbf{d}' = (\mathbf{R}\mathbf{x} + \mathbf{t}) - \mathbf{x}_0, \mathbf{n}' = \mathbf{R}\mathbf{n}, \mathbf{v}' = \mathbf{R}\mathbf{v}. \quad (4)$$

This process actually aligns a newly deformed mesh state $\mathbf{x}$ with its rest state $\mathbf{x}_0$.

3 Kinematics-based loss function

We minimize a kinematics-based loss to constrain the relationship between the synthesized velocities and positions, as

$$\mathcal{L}_{kine} = \sum_{k=1}^{n} ||(\mathbf{x}_k^s - (\mathbf{x}^s + (\sum_{j=1}^{k} \mathbf{v}_{k-j}^s) * \Delta t)||^2, \quad (5)$$

where n is the length of frames associated to the input frame, and Δt represents the time step between consecutive frames. The $\mathbf{x}$ in this formula can be replaced by displacement. Moreover, we add a rigid motion transformation to displacement and velocity in the equation (4). Replace the $\mathbf{x}$ and $\mathbf{v}$ in the formula (5) by equation (4), the kinematics-based loss can be written as

$$\mathcal{L}_{kine} = \sum_{k=1}^{n} ||\mathbf{R}_k^{-1}(\mathbf{d'}_k^s - (\mathbf{d'}^s + (\sum_{j=1}^{k} \mathbf{v'}_{k-j}^s) * \Delta t)||^2, \quad (6)$$

Since the time step is really small and the the neighbouring n frames of meshes only have small scale deformation, we share the rotation matrix in (6) and get

$$\mathcal{L}_{kine} = \sum_{k=1}^{n} ||\mathbf{R}^{-1}(\mathbf{d'}_k^s - (\mathbf{d'}^s + (\sum_{j=1}^{k} \mathbf{v'}_{k-j}^s) * \Delta t)||^2. \quad (7)$$

To define the notifications in an easily understood manner, we use the $(\mathbf{d}, \mathbf{n}, \mathbf{v})$ to replace the $(\mathbf{d}', \mathbf{n}', \mathbf{v}')$ in our paper.

References

[1] Liang J, Lin M C. Machine learning for digital try-on: Challenges and progress. *Computational Visual Media*, 2020, 7(2):159–167. DOI: 10.1007/s41095-020-0189-1

[2] Wang M, Lyu X Q, Li Y J, Zhang F L. VR content creation and exploration with deep learning: A survey. *Computational Visual Media*, 2020, 6(1):3–28. DOI: 10.1007/s41095-020-0162-z

[3] Terzopoulos D, Platt J, Barr A, Fleischer K. Elastically deformable models. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques - SIGGRAPH '87*, 1987, pp. 205–214. DOI: 10.1145/37401.37427

[4] Provot X. Collision and self-collision handling in cloth model dedicated to design garments. In *EG Workshop on Computer Animation and Simulation*, 1997, pp. 177–189. DOI: 10.1007/978-3-7091-6874-5_13

[5] Baraff D, Witkin A. Large steps in cloth simulation. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques - SIGGRAPH '98*, 1998, pp. 43–54. DOI: 10.1145/280814.280821

[6] Bridson R, Marino S, Fedkiw R. Simulation of clothing with folds and wrinkles. In *ACM SIGGRAPH 2005 Courses on - SIGGRAPH '05*, 2005, pp. 28–36. DOI: 10.1145/1198555.1198573

[7] Wang H, Hecht F, Ramamoorthi R, O'Brien J F. Example-based wrinkle synthesis for clothing animation. *ACM Trans. Graph.*, 2010, 29(4):1–8. DOI: 10.1145/1778765.1778844

[8] Zurdo J, Brito J, Otaduy M. Animating wrinkles by example on non-skinned cloth. *IEEE Trans. Visual. Comput. Graphics*, 2013, 19(1):149–158. DOI: 10.1109/tvcg.2012.79

[9] Santesteban I, Otaduy M A, Casas D. Learning-based animation of clothing for virtual try-on. In *Computer Graphics Forum*, volume 38, 2019, pp. 355–366. DOI: 10.1111/cgf.13643

[10] Feng W W, Yu Y, Kim B U. A deformation transformer for real-time cloth animation. *ACM Trans. Graph.*, 2010, 29(4):1–9. DOI: 10.1145/1778765.1778845

[11] Aguiar E, Sigal L, Treuille A, Hodgins J K. Stable spaces for real-time clothing. *ACM Trans. Graph.*, 2010, 29(4):1–9. DOI: 10.1145/1778765.1778843

[12] Kavan L, Gerszewski D, Bargteil A W, Sloan P P. Physics-inspired upsampling for cloth simulation in games. *ACM Trans. Graph.*, 2011, 30(4):1–10. DOI: 10.1145/2010324.1964988

[13] Chen L, Ye J, Jiang L, Ma C, Cheng Z, Zhang X. Synthesizing cloth wrinkles by CNN-based geometry image superresolution. *Comput Anim Virtual Worlds*, 2018, 29(3-4):e1810. DOI: 10.1002/cav.1810

[14] Oh Y J, Lee T M, Lee I K. Hierarchical cloth simulation using deep neural networks. In *Proceedings of Computer Graphics International 2018*, pp. 139–146. 2018. DOI: 10.1145/3208159.3208162

[15] Lähner Z, Cremers D, Tung T. DeepWrinkles: Accurate and realistic clothing modeling. In *European Conference on Computer Vision*, 2018, pp. 698–715. DOI: 10.1111/cgf.13643

[16] Ledig C, Theis L, Huszar F, Caballero J, Cunningham A, Acosta A, Aitken A, Tejani A, Totz J, Wang Z, Shi W. Photo-realistic single image super-resolution using a generative adversarial network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 105–114. DOI: 10.1109/cvpr.2017.19

[17] Zhang Y, Tian Y, Kong Y, Zhong B, Fu Y. Residual dense network for image super-resolution. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1–10. DOI: 10.1109/cvpr.2018.00262

[18] Gu X, Gortler S J, Hoppe H. Geometry images. *ACM Trans. Graph.*, 2002, 21(3):355–361. DOI: 10.1145/566654.566589

[19] Narain R, Samii A, O'Brien J F. Adaptive anisotropic remeshing for cloth simulation. *ACM Trans. Graph.*, 2012, 31(6):1–10. DOI: 10.1145/2366145.2366171

[20] Liu T, Bargteil A W, O'Brien J F, Kavan L. Fast simulation of mass-spring systems. *ACM Trans. Graph.*, 2013, 32(6):1–7. DOI: 10.1145/2508363.2508406

[21] Guan P, Reiss L, Hirshberg D A, Weiss A, Black M J. Drape. *ACM Trans. Graph.*, 2012, 31(4):1–10. DOI: 10.1145/2185520.2185531

[22] Kim D, Koh W, Narain R, Fatahalian K, Treuille A, O'Brien J F. Near-exhaustive precomputation of secondary cloth effects. *ACM Trans. Graph.*, 2013, 32(4):1–8. DOI: 10.1145/2461912.2462020

[23] Gundogdu E, Constantin V, Seifoddini A, Dang M, Salzmann M, Fua P. GarNet: A two-stream network for fast and accurate 3D cloth draping. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 8738–8747. DOI: 10.1109/iccv.2019.00883

[24] Wang T Y, Ceylan D, Popović J, Mitra N J. Learning a shared shape space for multimodal garment design. *ACM Trans. Graph.*, 2019, 37(6):1–13. DOI: 10.1145/3272127.3275074

[25] Wang T Y, Shao T, Fu K, Mitra N J. Learning an intrinsic garment space for interactive authoring of garment animation. *ACM Trans. Graph.*, 2019, 38(6):1–12. DOI: 10.1145/3355089.3356512

[26] Hahn F, Thomaszewski B, Coros S, Sumner R W, Cole F, Meyer M, DeRose T, Gross M. Subspace clothing simulation using adaptive bases. *ACM Trans. Graph.*, 2014, 33(4):1–9. DOI: 10.1145/2601097.2601160

[27] Xiao Y P, Lai Y K, Zhang F L, Li C, Gao L. A survey on deep geometry learning: From a representation perspective. *Computational Visual Media*, 2020, 6(2):113–133. DOI: 10.1007/s41095-020-0174-8

[28] Yuan Y J, Lai Y K, Wu T, Gao L, Liu L. A revisit of shape editing techniques: From the geometric to the neural viewpoint. *CoRR*, 2021, abs/2103.01694.

[29] Wang P S, Liu Y, Guo Y X, Sun C Y, Tong X. O-cnn: Octree-based convolutional neural networks for 3d shape analysis. *ACM Trans. Graph.*, 2017, 36(4):1–11. DOI: 10.1145/3072959.3073608

[30] Su H, Maji S, Kalogerakis E, Learned-Miller E. Multi-view convolutional neural networks for 3D shape recognition. In *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 945–953. DOI: 10.1109/iccv.2015.114

[31] Sinha A, Bai J, Ramani K. Deep learning 3D shape surfaces using geometry images. In *European Conference on Computer Vision (ECCV)*, 2016, pp. 223–240. DOI: 10.1007/978-3-319-46466-4_14

[32] Tan Q, Gao L, Lai Y, Yang J, Xia S. Mesh-based autoencoders for localized deformation component analysis. In McIlraith S A, Weinberger K Q, editors, *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18)*, 2018, pp. 2452–2459.

[33] Tan Q, Gao L, Lai Y K, Xia S. Variational autoencoders for deforming 3D mesh models. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5841–5850. DOI: 10.1109/cvpr.2018.00612

[34] Gao L, Lai Y K, Liang D, Chen S Y, Xia S. Efficient and flexible deformation representation for data-driven surface modeling. *ACM Trans. Graph.*, 2016, 35(5):1–17. DOI: 10.1145/2908736

[35] Gao L, Lai Y K, Yang J, Zhang L X, Xia S, Kobbelt L. Sparse data driven mesh deformation. *IEEE Trans. Visual. Comput. Graphics*, 2021, 27(3):2085–2100. DOI: 10.1109/tvcg.2019.2941200

[36] Zhang M, Wang T, Ceylan D, Mitra N J. Deep detail enhancement for any garment. *arXiv preprint arXiv:2008.04367*, 2020.

[37] Dong C, Loy C C, He K, Tang X. Image super-resolution using deep convolutional networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2016, 38(2):295–307. DOI: 10.1109/tpami.2015.2439281

[38] Liu S, Gang R, Li C, Song R. Adaptive deep residual network for single image super-resolution. *Computational Visual Media*, 2019, 5(4):391–401. DOI: 10.1007/s41095-019-0158-8

[39] Yue H J, Shen S, Yang J Y, Hu H F, Chen Y F. Reference image guided super-resolution via progressive channel attention networks. *J. Comput. Sci. Technol.*, 2020, 35(3):551–563. DOI: 10.1007/s11390-020-0270-3

[40] Dong C, Loy C C, Tang X. Accelerating the super-resolution convolutional neural network. In *European Conference on Computer Vision*, 2016, pp. 391–407. DOI: 10.1007/978-3-319-46475-6_25

[41] Shi W, Caballero J, Huszar F, Totz J, Aitken A P, Bishop R, Rueckert D, Wang Z. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 1874–1883. DOI: 10.1109/cvpr.2016.207

[42] Haris M, Shakhnarovich G, Ukita N. Deep back-projection networks for super-resolution. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1664–1673. DOI: 10.1109/cvpr.2018.00179

[43] Kappeler A, Yoo S, Dai Q, Katsaggelos A K. Video super-resolution with convolutional neural networks. *IEEE Trans. Comput. Imaging*, 2016, 2(2):109–122. DOI: 10.1109/tci.2016.2532323

[44] Chu M, Xie Y, Mayer J, Leal-Taixé L, Thuerey N. Learning temporal coherence via self-supervision for GAN-based video generation. *ACM Trans. Graph.*, 2020, 39(4):75–1. DOI: 10.1145/3386569.3392457

[45] Bhattacharjee P, Das S. Directional attention based video frame prediction using graph convolutional networks. In *2019 International Joint Conference on Neural Networks (IJCNN)*, 2019, pp. 4268–4277. DOI: 10.1109/ijcnn.2019.8852090

[46] Xie Y, Franz E, Chu M, Thuerey N. tempogan: A temporally coherent, volumetric gan for super-resolution fluid flow. *ACM Trans. Graph.*, 2018, 37(4):1–15. DOI: 10.1145/3197517.3201304

[47] Kabsch W. A discussion of the solution for the best rotation to relate two sets of vectors. *Acta Cryst Sect A*, 1978, 34(5):827–828. DOI: 10.1107/s0567739478001680

[48] Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010, pp. 249–256.

[49] Bergou M, Mathur S, Wardetzky M, Grinspun E. TRACKS: Toward directable thin shells. *ACM Trans. Graph.*, 2007, 26(3):50. DOI: 10.1145/1276377.1276439

[50] Müller M, Gross M. Interactive virtual materials. In *Proceedings of Graphics Interface 2004*, 2004, pp. 239–246.

[51] Caruana R. Multitask learning. *Mach. Learn.*, 1997, 28(1):41–75. DOI: 10.1023/a:1007379606734

[52] Burden R L, Faires J D. *Numerical analysis (9th edition)*. Cengage Learning, 2011.

[53] Ye J, Ma G, Jiang L, Chen L, Li J, Xiong G, Zhang X, Tang M. A unified cloth untangling framework through discrete collision detection. *Comput. Graph. Forum*, 2017, 36(7):217–228. DOI: 10.1111/cgf.13287

[54] Wang H, O'Brien J F, Ramamoorthi R. Data-driven elastic models for cloth. *ACM Trans. Graph.*, 2011, 30(4):1–12. DOI: 10.1145/2010324.1964966

[55] Kingma D P, Ba J. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations*, 2015, pp. 1–15.