Deep Physics-aware Inference of Cloth Deformation for Monocular Human Performance Capture – Supplemental Document –

Yue Li[1,2,3], Marc Habermann[1,2], Bernhard Thomaszewski[3], Stelian Coros[3], Thabo Beeler[4], and Christian Theobalt[1,2]

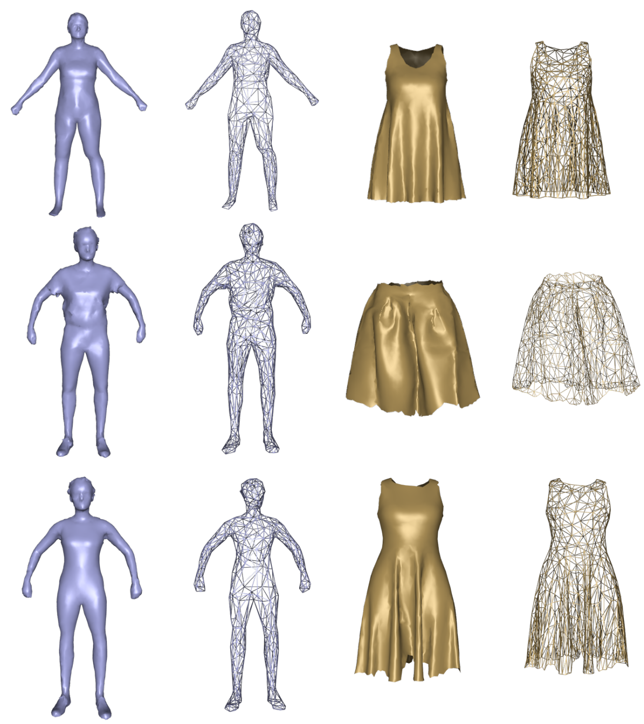
[1]Max Planck Institute for Informatics, [2]Saarland Informatics Campus, [3]ETH Zurich, [4]Google Inc.

In the following, we first provide more details about the template processing (Sec. 1) where we define the rest shape for our simulation and give illustrations of our templates and embedded graphs. Then, we explain in more detail regarding different simulation components, including the cloth model, the time integration scheme, collision resolution, and failure case handling (Sec. 2). Finally, we provide the weights used for the learning objectives and the barycentric weights formulation for the attachment loss (Sec. 3).

1. Template Processing

Rest Shape. We find our cloth rest shape by isometrically mapping all triangles of the template onto a 2D plane. Specifically, we move one vertex of each triangle to the origin, align one edge that shares this vertex to the x-axis, and rotate the third vertex into the xy-plane.

Templates and Embedded Graphs. In Fig. 1, we visualize our separated templates as well as the corresponding graph meshes for all of our subjects. Note that the estimated naked body looks plausible and can thus be used in our simulation layer as a collision proxy.



Body Mesh Body Graph Cloth Mesh Cloth Graph

Figure 1. Template and graph meshes of our subjects. The 3D body geometries are plausible and can thus be used to drive our cloth simulation.

2. Simulation Details

We adopt the finite element-based simulation framework ARCSim [7] and employ an optimization-based implicit Euler method [6] for time integration. We use Continuous Collision Detection [8] together with the method by Harmon et al. [5] for collision response.

Cloth Model. We adopt the same cloth model used by Thomaszewski *et al*. [9], where the in-plane stretching behavior is defined on a triangle basis using nonlinear constant strain triangles [1] and the out-of-plane bending force is computed between adjacent triangle pairs adapted from Bridson *et al*. [3]. Here we provide the detailed formulation in our implementation.

We use an isotropic St.Venant-Kirchhoff constitutive model [2] to capture the in-plane deformations. Given the deformed 3D vertex locations ($\mathbf{x}$) of a triangle and its undeformed rest configuration in 2D ($\mathbf{X}$), we compute the triangle-wise stretching energy as

$$E_{\text{stretch}} = \mu||\mathbf{E}||^2 + \frac{\lambda}{2}tr(\mathbf{E})^2 \tag{1}$$

where $tr(\cdot)$ is the trace operator, and the non-linear Green strain $\mathbf{E}$ is defined as $\mathbf{E} = \mathbf{F}^T\mathbf{F} - \mathbf{I}$, and $\mathbf{F} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}}$ is the deformation gradient of a given triangle. μ and λ are Lamé parameters which are computed from Young's modulus Y and Poisson ratio ν as

$$\mu = \frac{Y}{2(1+\nu)}, \lambda = \frac{Y\nu}{(1+\nu)(1-2\nu)}. \quad (2)$$

We obtain $\mathbf{X}$ from the template as described in Sec. 1.

Bending energy is defined on the common edge ($\mathbf{e}$) of adjacent triangle pairs, which reads

$$E_{\text{bend}} = \frac{3}{2} k_{\text{bend}} \frac{|\mathbf{e}|^2}{|\mathbf{n_1}| + |\mathbf{n_2}|} (\theta - \theta^0)^2 \quad (3)$$

where θ, θ^0 are the dihedral angles of an adjacent triangle pair in the deformed and undeformed configurations, k_{bend} is the bending stiffness, $\mathbf{n_1}, \mathbf{n_2}$ are the normal vectors of the neighboring triangles. It is worth noting that this model has only three parameters and these are intuitive to adjust. We use $k_{\text{bend}} = 10^{-6}, Y = 10^5$ and $\mu = 0.42$ for all of our simulations.

Time Integration. To allow stable simulations with step sizes that match the time-stepping from the captured data, we rely on implicit integration. Specifically, we use the first-order implicit Euler method, whose update rules solve the minimization problem

$$\min_{\mathbf{x}} \frac{1}{2}(\mathbf{x}-\mathbf{y})^T \mathbf{M}(\mathbf{x}-\mathbf{y}) + \Delta t^2 E_{\text{pot}}(\mathbf{x}) , \quad (4)$$

where $E_{\text{pot}} = E_{\text{int}} + E_{\text{ext}} + E_{\text{cons}} + E_{\text{col}}$ collects all potential energy contributions, $\mathbf{x}$ are end-of-time-step positions, and $\mathbf{y} = \mathbf{x}^n - \mathbf{v}^n \Delta t$ are first-order predictions for positions. Δt is the time step, which we choose to be the frame rate of the input video sequence.

Collision Resolution. Given the solution $\mathbf{x}_{\text{IE}}$ from Eq. 4, we seek for a collision-free state $\mathbf{x}$. We perform Continuous Collision Detection first and group the intersecting edge-edge (EE), vertex-face (VF) pairs into impact zones [5]. We then resolve collisions by solving the following constraint minimization problem [7].

$$\begin{aligned} \min_{\mathbf{x}} \frac{1}{2}(\mathbf{x}-\mathbf{x}_{\text{IE}})^T \tilde{\mathbf{M}}(\mathbf{x}-\mathbf{x}_{\text{IE}}) \\ s.t. \mathbf{G}\mathbf{x} \leq \mathbf{h} \end{aligned} \quad (5)$$

where $\tilde{\mathbf{M}}$ is a diagonal mass matrix divided by the sum of the mass of all vertices and the constraints take the form of

$$\mathbf{n} \sum_{i=0}^{4} \omega_i \mathbf{x}_i \leq d \quad (6)$$

where the $\mathbf{x}$ are the four vertices involved in an EE/VF collision pair. Given a VF pair, $\mathbf{n}$ is the face normal. Otherwise, if an EE pair is given, $\mathbf{n}$ is the cross product of the two edges. d is a user-defined distance threshold. Eq. 6 indicates that the tetrahedron which is formed by these four vertices should have positive volume.

Failure Case Handling. The simulation can fail for certain configurations, *e.g.* when a piece of cloth lies inside of self-penetrating body geometry. In this case, the collision resolving is not able to find a solution. We exit the simulation when the objective function is above a threshold and ignore the loss and gradient for this frame. We further initialize the next frame velocity from the forward differences of the network outputs again if the simulation exits before the end of the $\mathcal{F}$ frames we defined. We choose $\mathcal{F} = 5$ in our implementation.

3. Additional Information

Weights. We adapt DeepCap [4]'s weighting scheme for the same terms. Additionally, we use $\beta_{\text{att}} = 0.2$, and $\beta_{\text{sim}} = 20,000$ for the attachment loss and simulation loss, respectively.

Barycentric Weights. Here we provide the formula for the barycentric weights $\boldsymbol{\gamma}$ in the attachment loss. Given the vertices $\mathbf{C_0}, \mathbf{C_1}, \mathbf{C_2}$ of the triangle, the barycentric weights of a given point $\mathbf{p}$ w.r.t. the three vertices are computed as

$$\gamma_0 = \frac{A(C_0PC_2)}{A(C_0C_1C_2)}, \gamma_1 = \frac{A(C_0PC_1)}{A(C_0C_1C_2)}, \gamma_2 = \frac{A(C_1PC_2)}{A(C_0C_1C_2)}, \quad (7)$$

where $A(\cdot)$ compute the area of a given triangle.

References

[1] Klaus-Jurgen Bathe. Finite element procedures prentice-hall. *New Jersey*, 1037, 1996. 1

[2] Javier Bonet and Richard D Wood. *Nonlinear continuum mechanics for finite element analysis*. Cambridge university press, 1997. 1

[3] Robert Bridson, Sebastian Marino, and Ronald Fedkiw. Simulation of clothing with folds and wrinkles. In *ACM SIGGRAPH 2005 Courses*, pages 3–es. 2005. 1

[4] Marc Habermann, Weipeng Xu, Michael Zollhofer, Gerard Pons-Moll, and Christian Theobalt. Deepcap: Monocular human performance capture using weak supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5052–5063, 2020. 2

[5] David Harmon, Etienne Vouga, Rasmus Tamstorf, and Eitan Grinspun. Robust treatment of simultaneous collisions. In *ACM SIGGRAPH 2008 papers*, pages 1–4. 2008. 1, 2

[6] Sebastian Martin, Bernhard Thomaszewski, Eitan Grinspun, and Markus Gross. Example-based elastic materials. *ACM Trans. Graph.*, 30(4), July 2011. 1

[7] Rahul Narain, Armin Samii, and James F O'brien. Adaptive anisotropic remeshing for cloth simulation. *ACM transactions on graphics (TOG)*, 31(6):1–10, 2012. 1, 2

[8] Xavier Provot. Collision and self-collision handling in cloth model dedicated to design garments. In *Computer Animation and Simulation'97*, pages 177–189. Springer, 1997. 1

[9] Bernhard Thomaszewski, Simon Pabst, and Wolfgang Straßer. Asynchronous cloth simulation. In *Computer Graphics International*, volume 2, page 2, 2008. 1