SIGGRAPH ASIA 2022 DAEGU

Neural Cloth Simulation

Hugo Bertiche, Meysam Madadi and Sergio Escalera

Universitat de Barcelona and Computer Vision Center

sa2022.siggraph.org

Motivation

Contributions

- Unsupervised Cloth Dynamics
- Disentangled Cloth Subspace
- Domain Analysis

Problem Definition

Cloth solver f:

$$x_t = f(\theta_t, x_{t-1}, v_{t-1}) = f(\theta_t, x_{t-1}, x_{t-2}) \quad (1)$$

Problem Definition

Cloth solver f:

$$x_t = f(\theta_t, x_{t-1}, v_{t-1}) = f(\theta_t, x_{t-1}, x_{t-2}) \tag{1}$$

Then,

$$\begin{aligned} x_{t-1} &= f(\theta_{t-1}, x_{t-2}, x_{t-3}) \\ x_{t-2} &= f(\theta_{t-2}, x_{t-3}, x_{t-4}) \\ &\dots \end{aligned} \tag{2}$$

Problem Definition

Cloth solver f:

$$x_t = f(\theta_t, x_{t-1}, v_{t-1}) = f(\theta_t, x_{t-1}, x_{t-2}) \tag{1}$$

Then,

$$\begin{aligned} x_{t-1} &= f(\theta_{t-1}, x_{t-2}, x_{t-3}) \\ x_{t-2} &= f(\theta_{t-2}, x_{t-3}, x_{t-4}) \\ &\dots \end{aligned} \tag{2}$$

Therfore:

$$\begin{aligned} x_t &= \hat{f}(\theta_t, \theta_{t-1}, \theta_{t-2}, \dots, \theta_0, x_0) \\ \Theta_t &= \{\theta_t, \theta_{t-1}, \theta_{t-2}, \dots, \theta_0\} \\ x_t &= \hat{f}(\Theta_t, x_0) \end{aligned} \tag{3}$$

Static vs Dynamic

Static

$$\theta_t = \theta_{t-1} = \theta_{t-2} = \ldots = \theta_0$$

Dynamic

Cloth Subspace Physics

We established:

$$\hat{f} : \Theta_t \to x_t \tag{4}$$

Cloth Subspace Physics

We established:

$$\hat{f} : \Theta_t \rightarrow x_t \tag{4}$$

Therefore, exists $\mathcal{Z} = \{z\} \subset \mathbb{R}^d$, such that:

$$\Theta_t \rightarrow z_t \rightarrow x_t \tag{5}$$

Cloth Subspace Physics

We established:

$$\hat{f} : \Theta_t \to x_t \tag{4}$$

Therefore, exists $\mathcal{Z} = \{z\} \subset \mathbb{R}^d$, such that:

$$\Theta_t \to z_t \to x_t \tag{5}$$

Neural Cloth Subspace Solver:

$$z_t = g(\theta_t, z_{t-1}) \tag{6}$$

Cloth Subspace Physics

We established:

$$\hat{f} : \Theta_t \rightarrow x_t \tag{4}$$

Therefore, exists $\mathcal{Z} = \{z\} \subset \mathbb{R}^d$, such that:

$$\Theta_t \rightarrow z_t \rightarrow x_t \tag{5}$$

Neural Cloth Subspace Solver:

$$z_t = g(\theta_t, z_{t-1}) \tag{6}$$

Such that:

$$z_t = z_{t-1} = \ldots = z_0 \leftrightarrow \theta_t = \theta_{t-1} = \ldots = \theta_0 \tag{7}$$

Neural Network

Suitable architecture:

- Encoder-Decoder
- Recurrent
- Static vs Dynamic behaviour

Descriptors

Static descriptors $\theta^S \in \mathbb{R}^{K \times 9}$:

- Joint relative orientations
- ~~Axis-angle~~
- ~~Quaternion~~
- 6D descriptors[1]
- Novel joint global orientation:

$$\hat{g}_j = R_j^{-1} g / ||g|| \tag{8}$$

[1] Zhou, Y., Barnes, C., Lu, J., Yang, J., & Li, H. (2019). On the continuity of rotation representations in neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 5745-5753).

Descriptors

Static descriptors $\theta^S \in \mathbb{R}^{K\times 9}$:

- Joint relative orientations
- ~~Axis-angle~~
- ~~Quaternion~~
- 6D descriptors[1]
- Novel joint global orientation:

$$\hat{g}_j = R_j^{-1} g / ||g|| \tag{8}$$

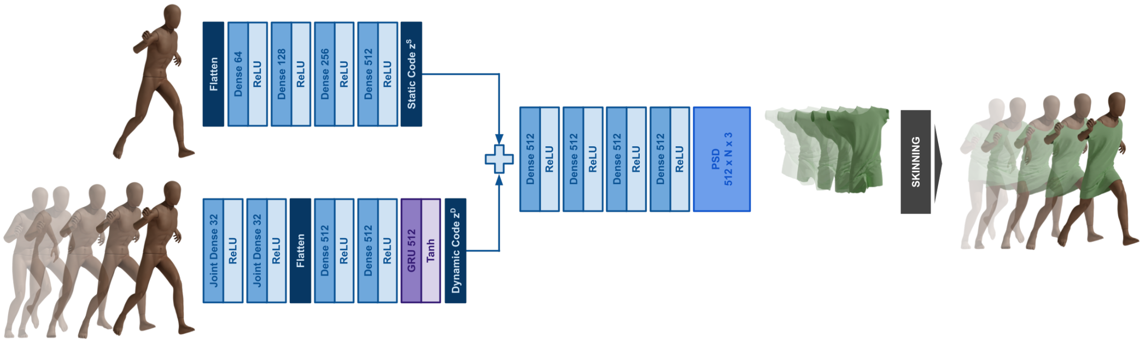
Dynamic descriptors $\theta^D \in \mathbb{R}^{K\times 12}$:

- Orientation velocity $\partial\theta^S/\partial t$
- Joint acceleration $\partial^2 J_x/\partial t^2$
- No motion $\leftrightarrow \theta^D = 0$
- *Unposed* accelerations:

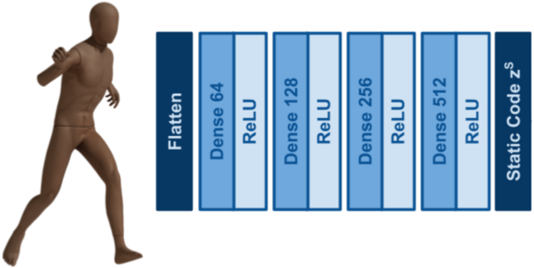
$$\hat{a}_j = R_j^{-1} a_j \tag{9}$$

[1] Zhou, Y., Barnes, C., Lu, J., Yang, J., & Li, H. (2019). On the continuity of rotation representations in neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 5745-5753).

Disentangled Network



Static Encoder



- Input: θ_t^S
- Output: static code z_t^S

Dynamic Encoder

- Input: $\{\theta_t^D, z_{t-1}^D\}$
- Output: dynamic code z_t^D
- No bias terms

Dynamic Encoder: Why No Bias?

- $z_t^D \propto \{\theta_t^D, z_{t-1}^D\}$
- $\theta_t^D = 0 \Rightarrow z_t^D \rightarrow 0$
- $\theta_t^D = ... = \theta_0^D = 0 \rightarrow z_t^D = 0$
- Invariant to 0-padding

Losses

Loss

$$\mathcal{L} = \mathcal{L}_{\text{cloth}} + \mathcal{L}_{\text{bending}} + L_{\text{collision}} + \mathcal{L}_{\text{inertia}} + \mathcal{L}_{\text{gravity}} = E(x; \theta) \tag{10}$$

- Loss = Energy
- Training = Simulating

Cloth Model

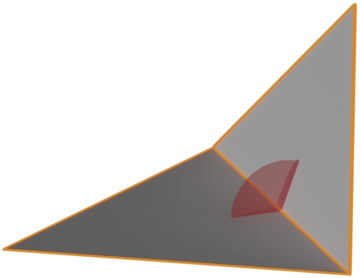
- Mass-spring
- Finite Element Methods
- Saint-Venant Kirchhoff
- ...

Bending

Bending Loss

$$\mathcal{L}_{\text{bending}} = k_b \frac{l^2}{8a}(\phi_t - \phi^R)^2 \qquad (11)$$

- k_b : bending stiffness
- l : edge length
- a : triangle area
- ϕ_t : dihedral angle at t
- ϕ^R : rest dihedral angle

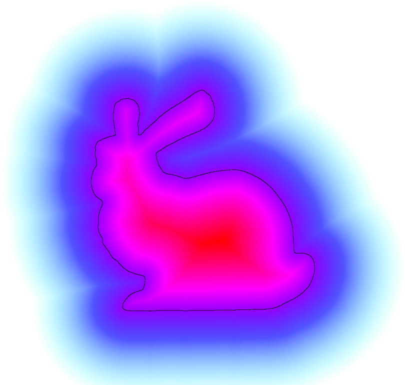


Collisions

Collision Loss

$$\mathcal{L}_{\text{collision}} = k_c \min(\text{d}(x_t;\theta_t) - \epsilon, 0)^2, \quad (12)$$

- k_c : collision *stiffness*
- $d(\cdot)$: signed distance
- x_t : cloth vertex locations at t
- θ_t : body parameterization
- ϵ : small threshold (robustness)



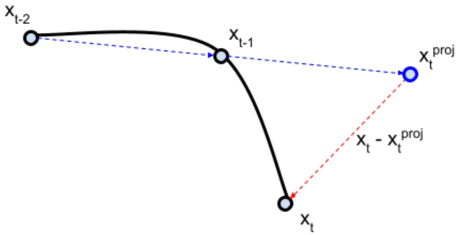
Inertia

Inertia Loss

$$\mathcal{L}_{\text{inertia}} = \frac{1}{2\Delta t^2} m(x_t - x_t^{\text{proj}})^2$$
$$x_t^{\text{proj}} = 2x_{t-1} - x_{t-2} \tag{13}$$

- Δt : time step
- m : vertex mass
- x_t : cloth vertex locations at t

Do NOT back-propagate through x_{t-1} and x_{t-2}!!!



Gravity

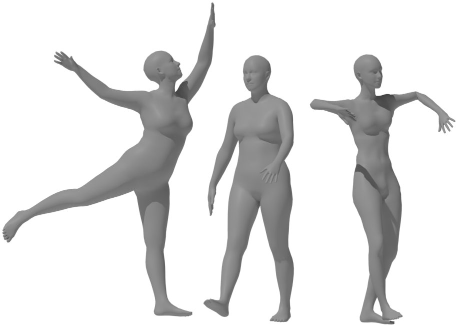
Gravity Loss

$$\mathcal{L}_{\text{gravity}} = -Mx_t g \tag{14}$$

- M : mass matrix
- x_t : vertex locations at t
- g : gravity

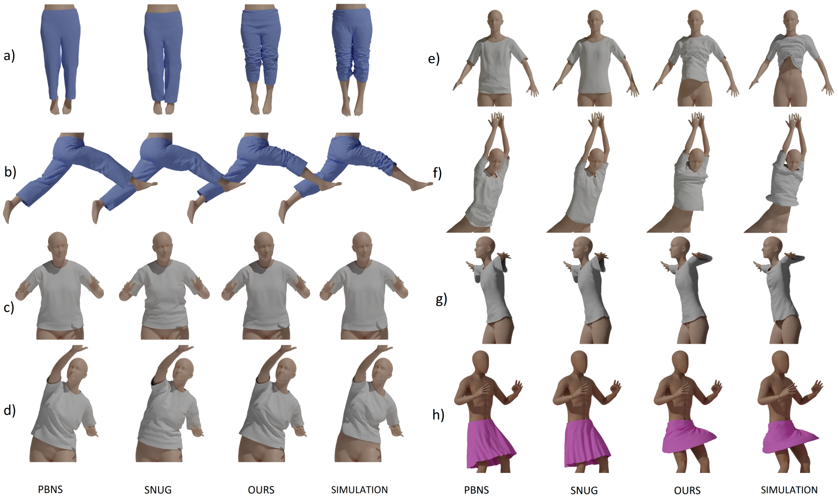
Data: Pose Sequences

Mixamo

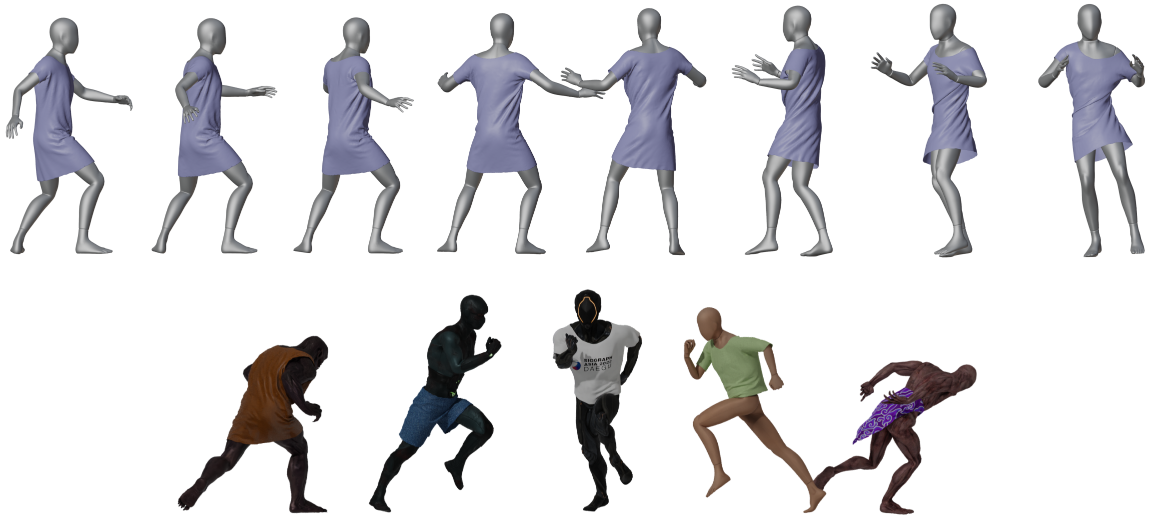


SMPL

PBNS vs SNUG vs NCS vs Simulation



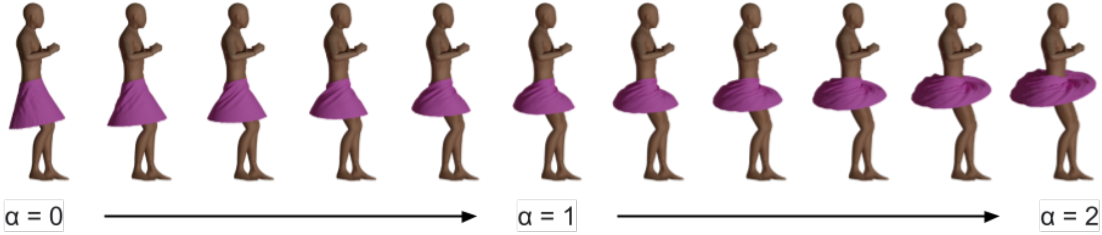
Qualitative



Motion Control

$$z = z^S + z^D \tag{15}$$

$$z_\alpha = z^S + \alpha z^D \tag{16}$$



Conclusions

- Unsupervised Cloth Dynamics
- Novel set of descriptors
- Disentangled subspaces for Static/Dynamic cloth deformations
    - Improved generalization
    - Novel motion augmentation (train)
    - Novel motion control (test)
- Compatible with arbitrary 3D characters/garments
- Real-time performance
- In-depth domain analysis

Thanks for your attention

Find out more:

- Project: `https://hbertiche.github.io/NeuralClothSim`
- Code: `https://github.com/hbertiche/NeuralClothSim`
- Video: `https://youtu.be/6HxXLBzRXFg`