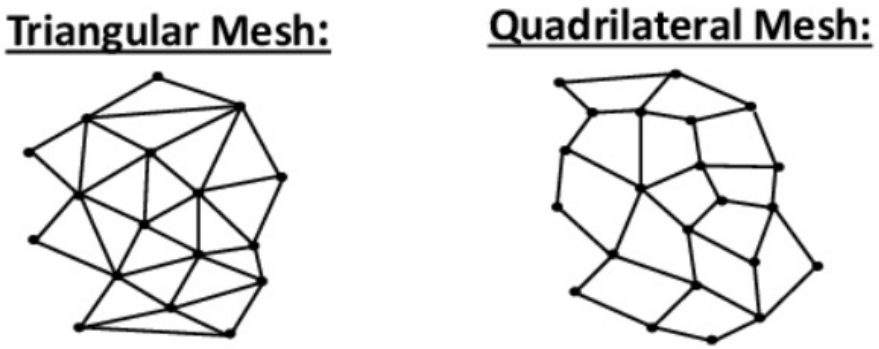
Instruction to SMPL and MANO

Jikai Wang

THE UNIVERSITY OF TEXAS AT DALLAS

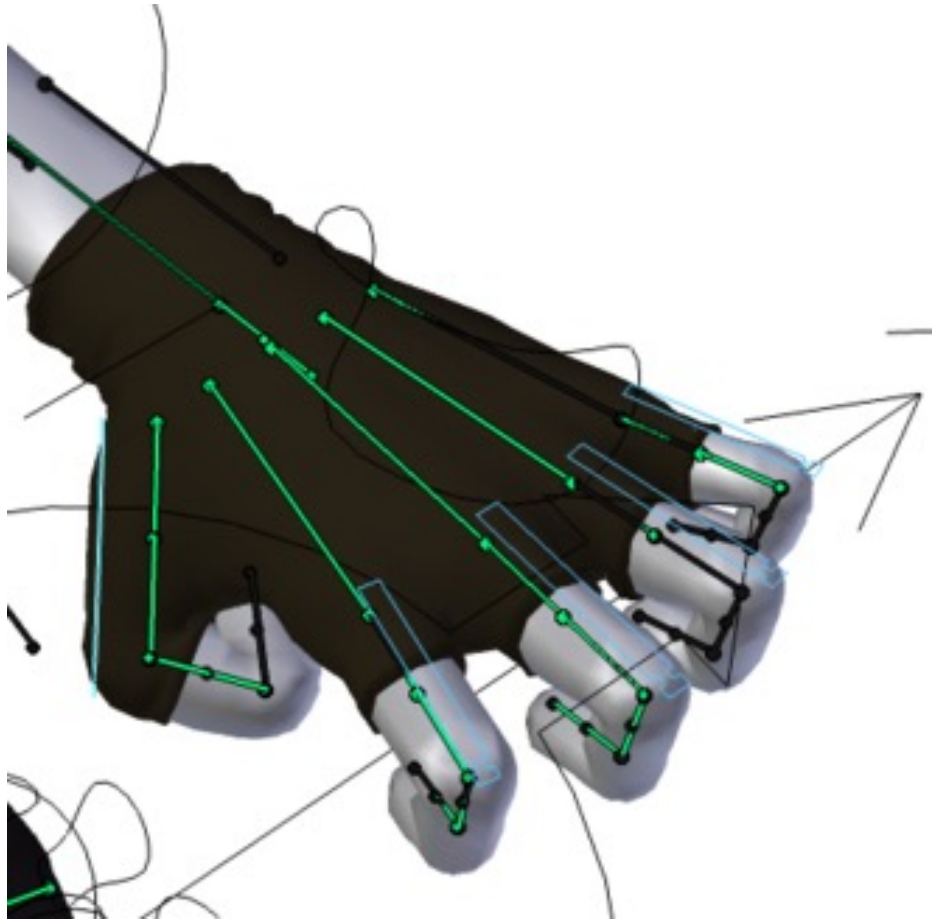
Basic Concepts

Basic Concepts



- Vertex
  - Position: 3D coordinate in space (x, y, z).
  - Color: the color information (r, g, b).
  - UV coordinate: used for texture mapping of the surface.
  - Normal Vector: used for lighting calculation, displacement mapping, etc.
  - Skinning Weights: weighting for assignment to bones to control animation
- Rigging
  - Mesh: a surface to represent the character (triangle mesh, quad mesh, etc.).
  - Bone (Skeleton): a hierarchical set of interconnected parts, which is a virtual representation used to animate the mesh.
    - one bone is associated with a group of vertices
    - one vertex could be assigned to multiple bones together with skinning weights
- Skinning Function: deform the mesh according to bones
  - Simple Skinning
  - Linear Blend Skinning (LBS)
  - Dual Quaternion Blend Skinning (DQBS)

Example of Rigging



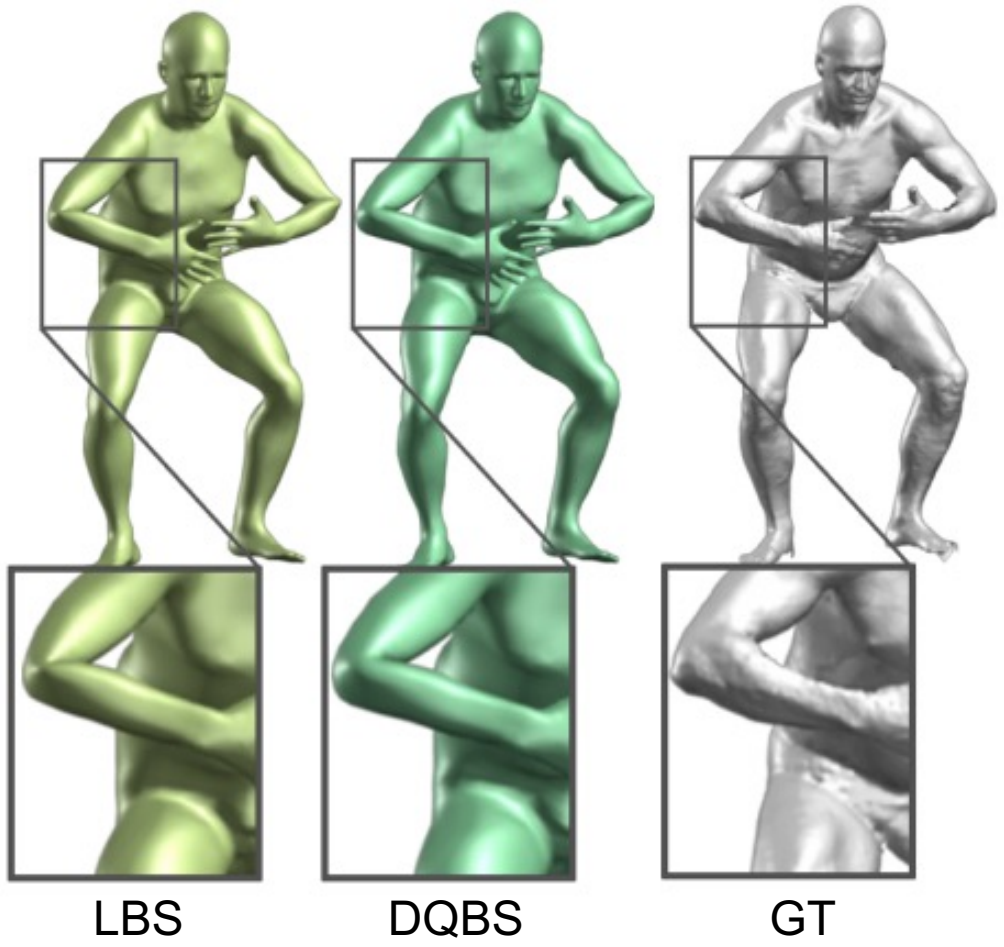
- Bones (in green) are used to pose a hand, they are virtual/invisible.
- By skinning function, we only need to pose the bones, and the mesh will be deformed accordingly.

THE UNIVERSITY OF TEXAS AT DALLAS

SMPL:
Skinned Multi-Person Linear Model

Why SMPL?

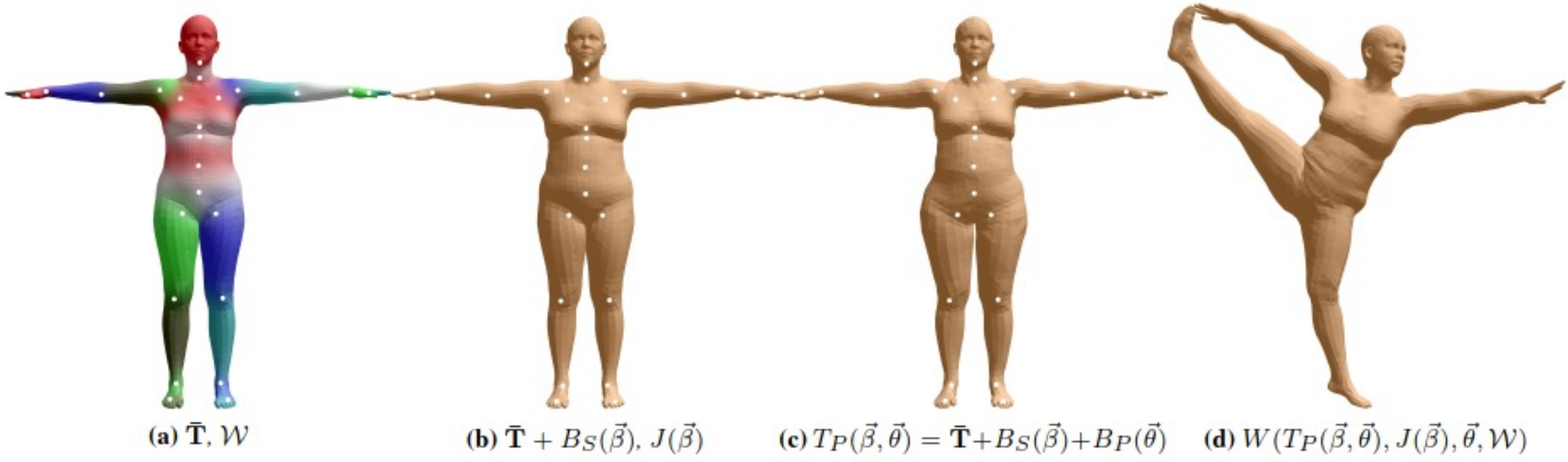
- Traditional methods just model the relationship between vertices and underlying skeleton structure.
- They may produce unrealistic deformations at joints.
- Need a realistic model representing human in different poses.



SMPL Model

- Mesh: $N = 6890$ vertices
- Bones: $K = 23$ joints
- Pose BlendShapes: controlled by vector θ
- Shape BlendShapes: controlled by vector β

SMPL Model



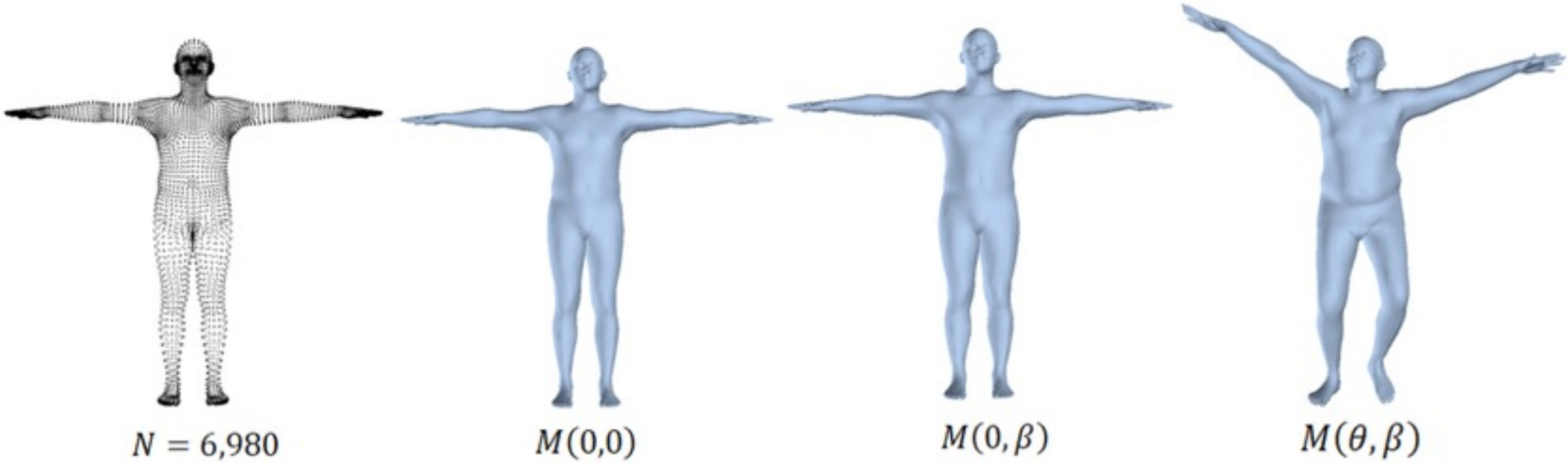
Template in T-pose

Shape Blendshapes added (Joints are regressed by $J(\beta)$)

Shape Blendshapes are added

Deform Mesh by Skinning Function

SMPL Model



SMPL - Functions

$M(\vec{\beta},\vec{\theta}) = W\left(T_P(\vec{\beta},\vec{\theta}), J(\vec{\beta}), \vec{\theta}, \mathcal{W}\right)$

$W(\overline{\mathbf{T}}, \mathbf{J}, \vec{\theta}, \mathcal{W}) : \mathbb{R}^{3N\times 3K\times|\vec{\theta}|\times|\mathcal{W}|} \mapsto \mathbb{R}^{3N}$ **(the standard linear blend skinning function)**

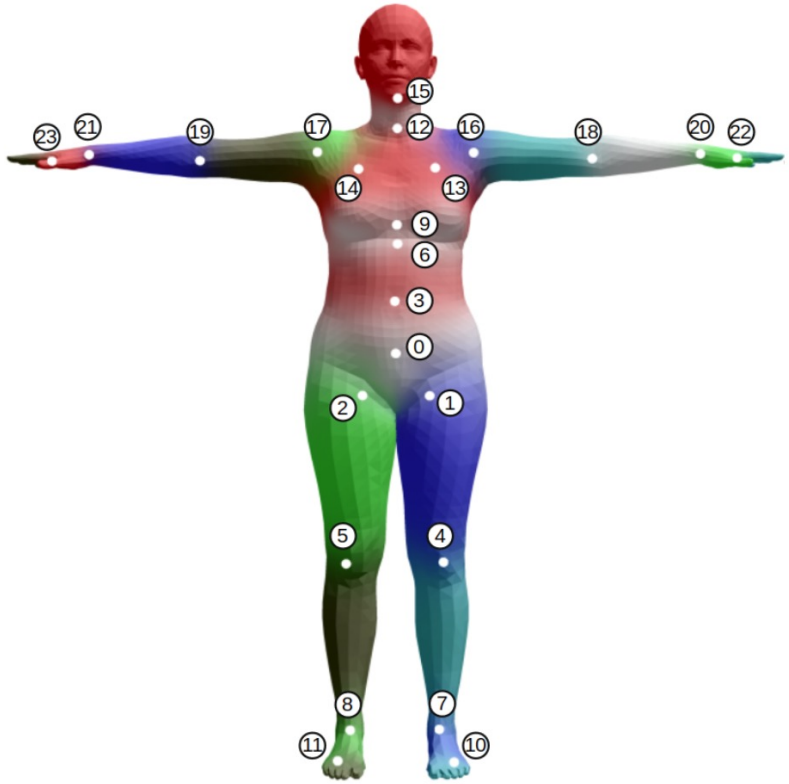
$J(\vec{\beta}) : \mathbb{R}^{|\vec{\beta}|} \mapsto \mathbb{R}^{3K}$:a function to predict K joint locations ;

$T_P(\vec{\beta},\vec{\theta}) = \overline{\mathbf{T}} + B_S(\vec{\beta}) + B_P(\vec{\theta})$

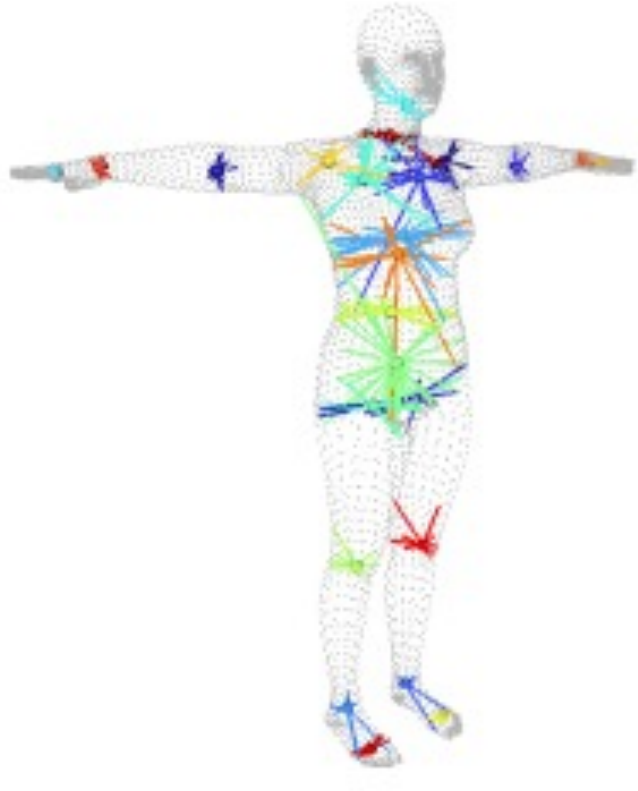
$B_S(\vec{\beta}) : \mathbb{R}^{|\vec{\beta}|} \mapsto \mathbb{R}^{3N}$: a blend shape function; input : shape parameters $\vec{\beta}$;output : a blend shape sculpting the subject identity

$B_P(\vec{\theta}) : \mathbb{R}^{|\theta|} \mapsto \mathbb{R}^{3N}$: a pose-dependent blend shape function ; $\vec{\theta}$: a vector of pose parameters , accounts for the effects of posedependent deformations

SMPL – Joint Regressor $J(\beta)$

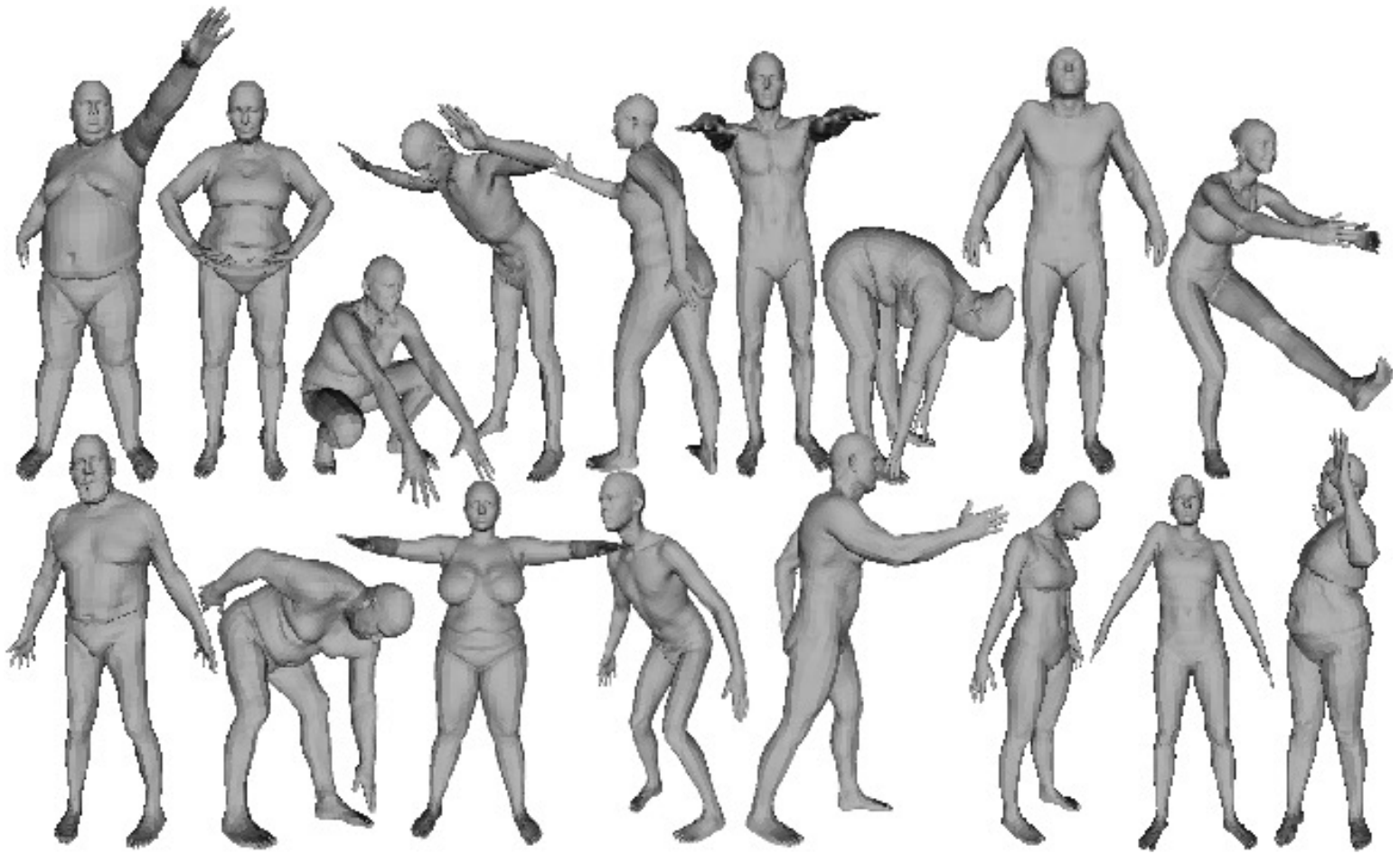


Initialization of Joints and Blend Weights



Joint Regression Function $J(\beta)$

SMPL



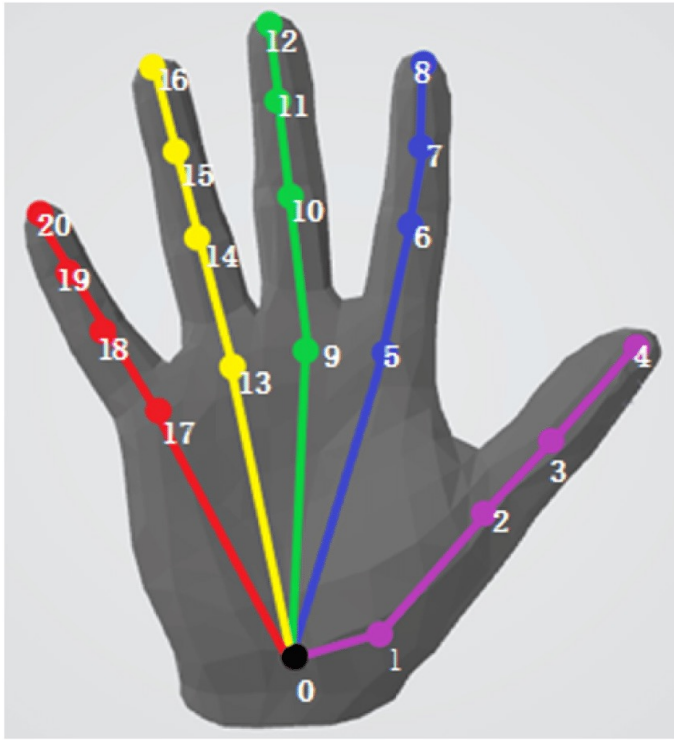
THE UNIVERSITY OF TEXAS AT DALLAS

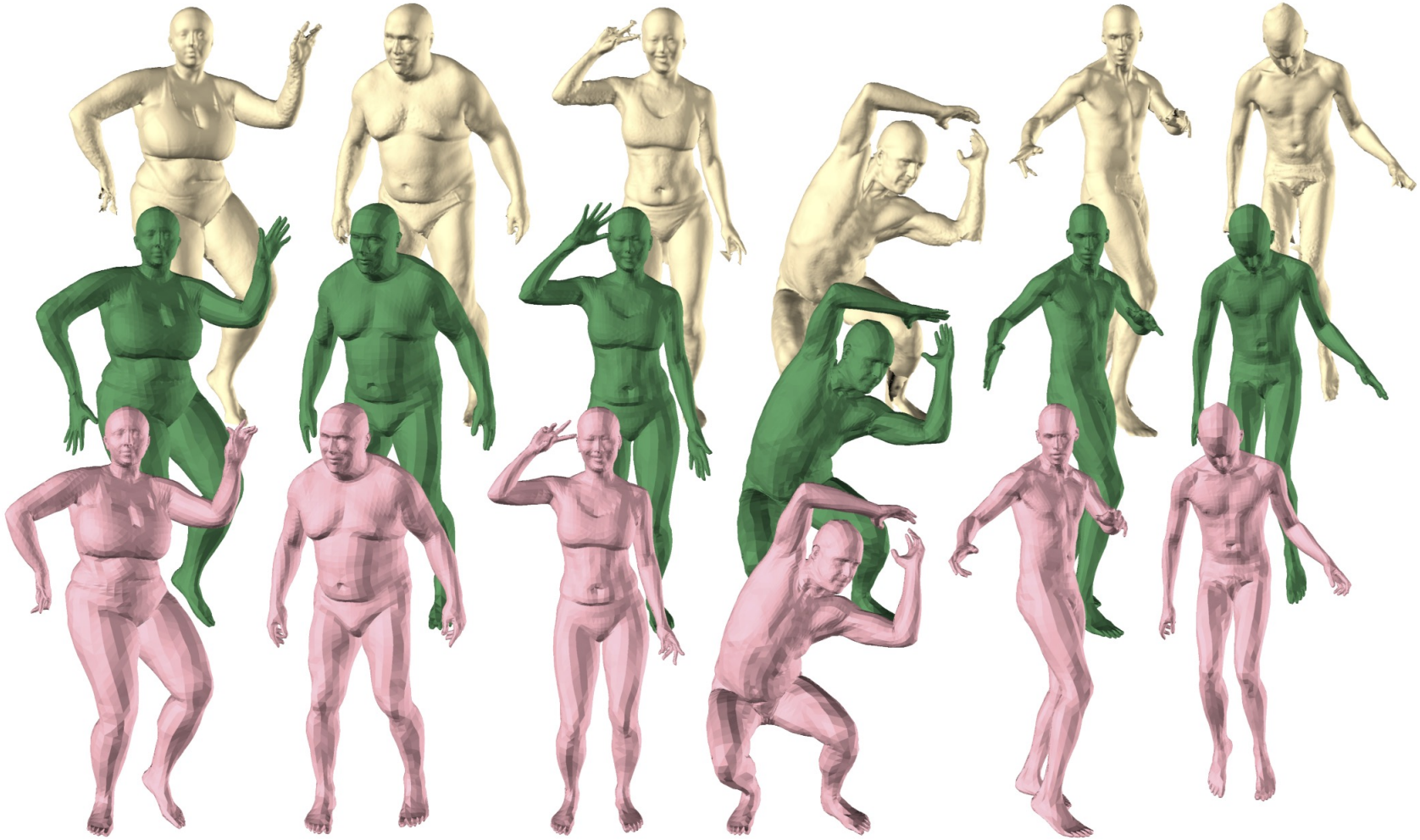
SMPL-H (MANO)

MANO

- MANO: a hand Model with Articulated and Non-rigid deformations
  - Based on SMPL model
  - Compatible with the full body SMPL model
- Steps:
  - Make the hand template: take the hand vertices from the SMPL model
  - Similar as SMPL, define joints, shape blendshapes, pose blendshapes for hand model
    - Bones: $K = 15$ joints + global orientation(wrist)
- Learn model parameters: (learn from right hand, and do a mirror for creating left hand)
  - The mean hand template in T-pose: T
  - Shape Blendshapes: β
  - Pose Blendshapes: θ
  - Joint Locations: $\mathrm{J}(\beta)$
  - Blending Weights: $\mathcal{W}$

SMPL+H





THE UNIVERSITY OF TEXAS AT DALLAS

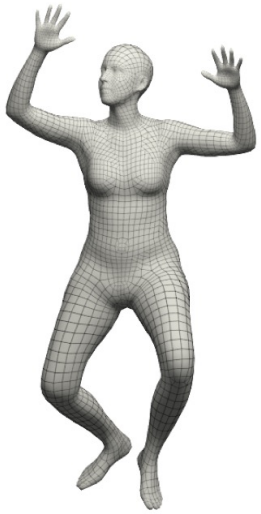
SMPL-X

SMPL-X

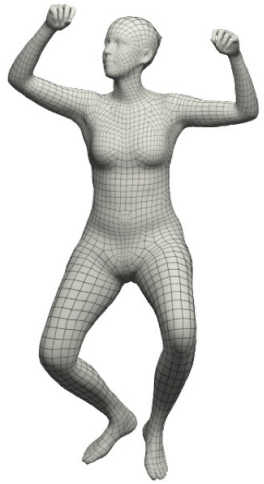
- SMPL-X: a more expressive human model
  - Realistic body: SMPL
  - Articulated hands: MANO
  - Expressive face
- Follows the formulation of SMPL

$$M(\beta, \theta, \psi) = W(T_p(\beta, \theta, \psi), J(\beta), \theta, \mathcal{W}) \in \mathbb{R}^{10475 \times 3}$$
$$T_P(\beta, \theta, \psi) = \bar{T} + B_S(\beta; \mathcal{S}) + B_E(\psi; \mathcal{E}) + B_P(\theta; \mathcal{P})$$

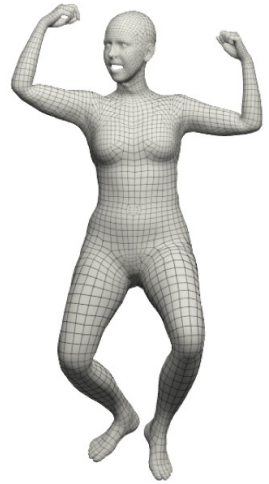
  - Body shape: β
  - Facial expression: ψ
  - Pose: $\theta = [\theta_b, \theta_h, \theta_f]$
    - θ_b: body pose
    - θ_h: hand pose
    - θ_f: jaw pose



SMPL



SMPL+H



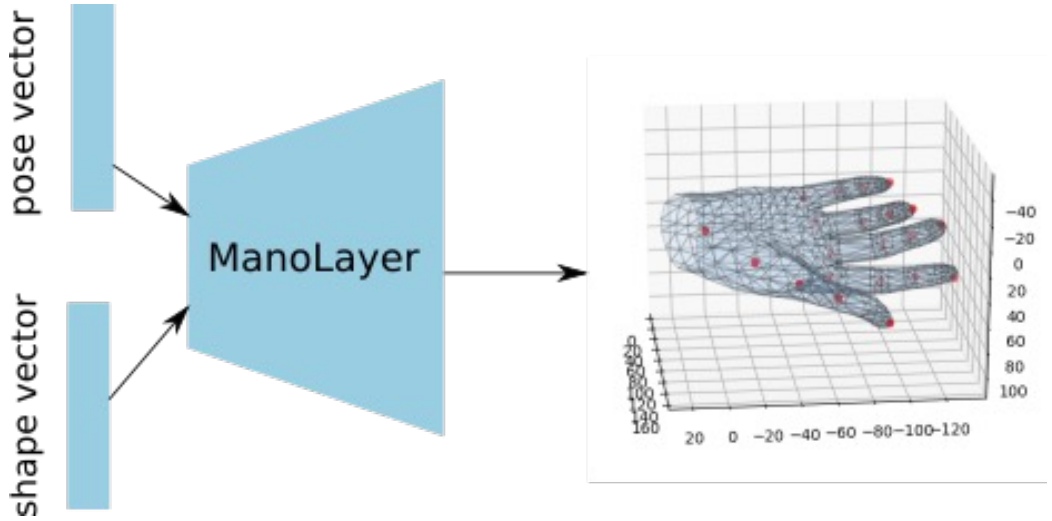
SMPL-X

THE UNIVERSITY OF TEXAS AT DALLAS

MANO Layer for PyTorch

MANO Layer for PyTorch (manopth)

- manopth
  - ManoLayer takes **batched** hand pose and shape vectors and outputs corresponding hand joints and vertices.



Resources

- Websites
  - SMPL
  - SMPL-H
  - SMPL-X
- manopth Github
  - manopth