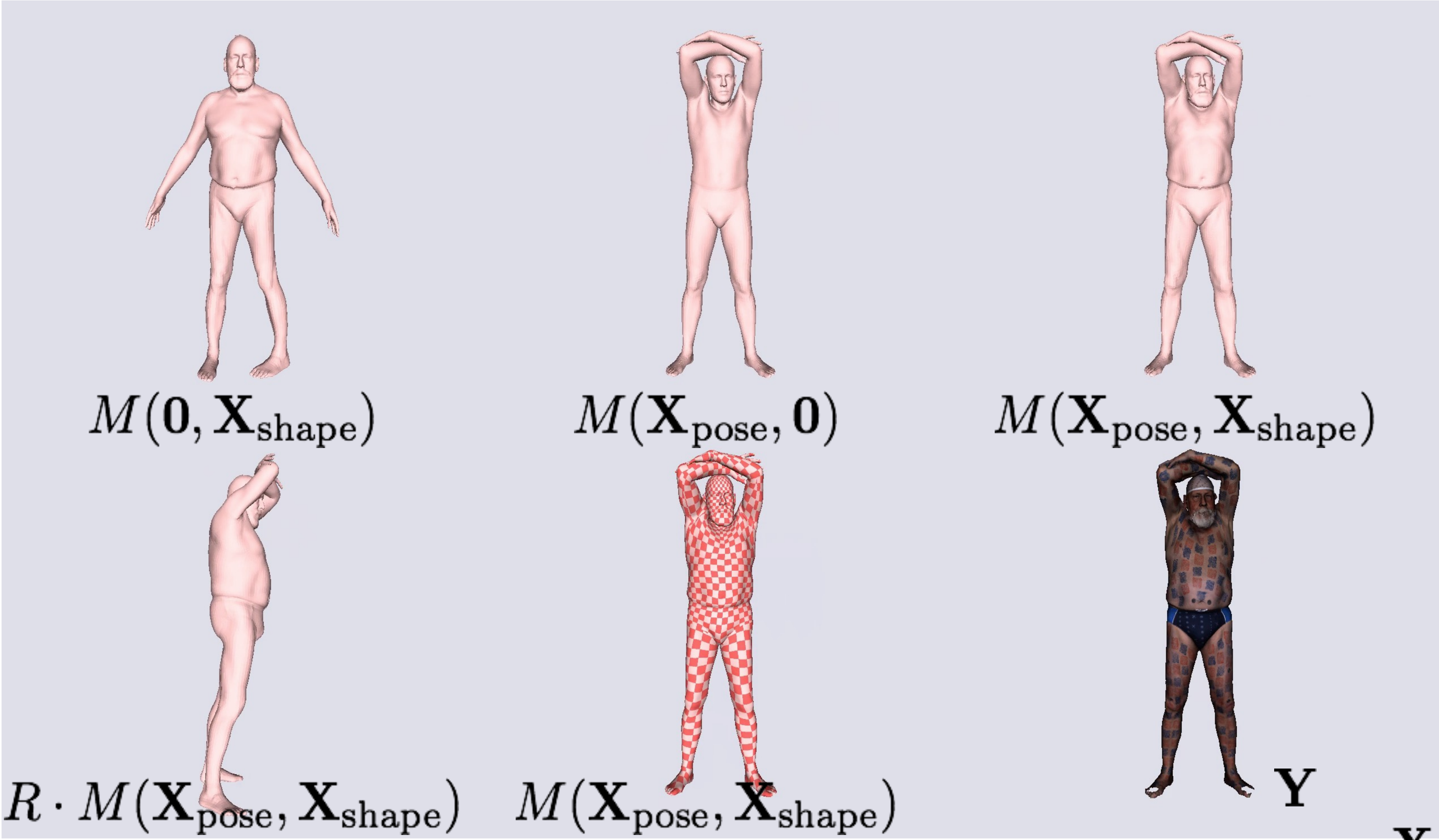
Virtual Humans – Winter 23/24

Lecture 4_1 – ICP: Vertex Based Models

Prof. Dr.-Ing. Gerard Pons-Moll
University of Tübingen / MPI-Informatics

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

A Body Model is a function

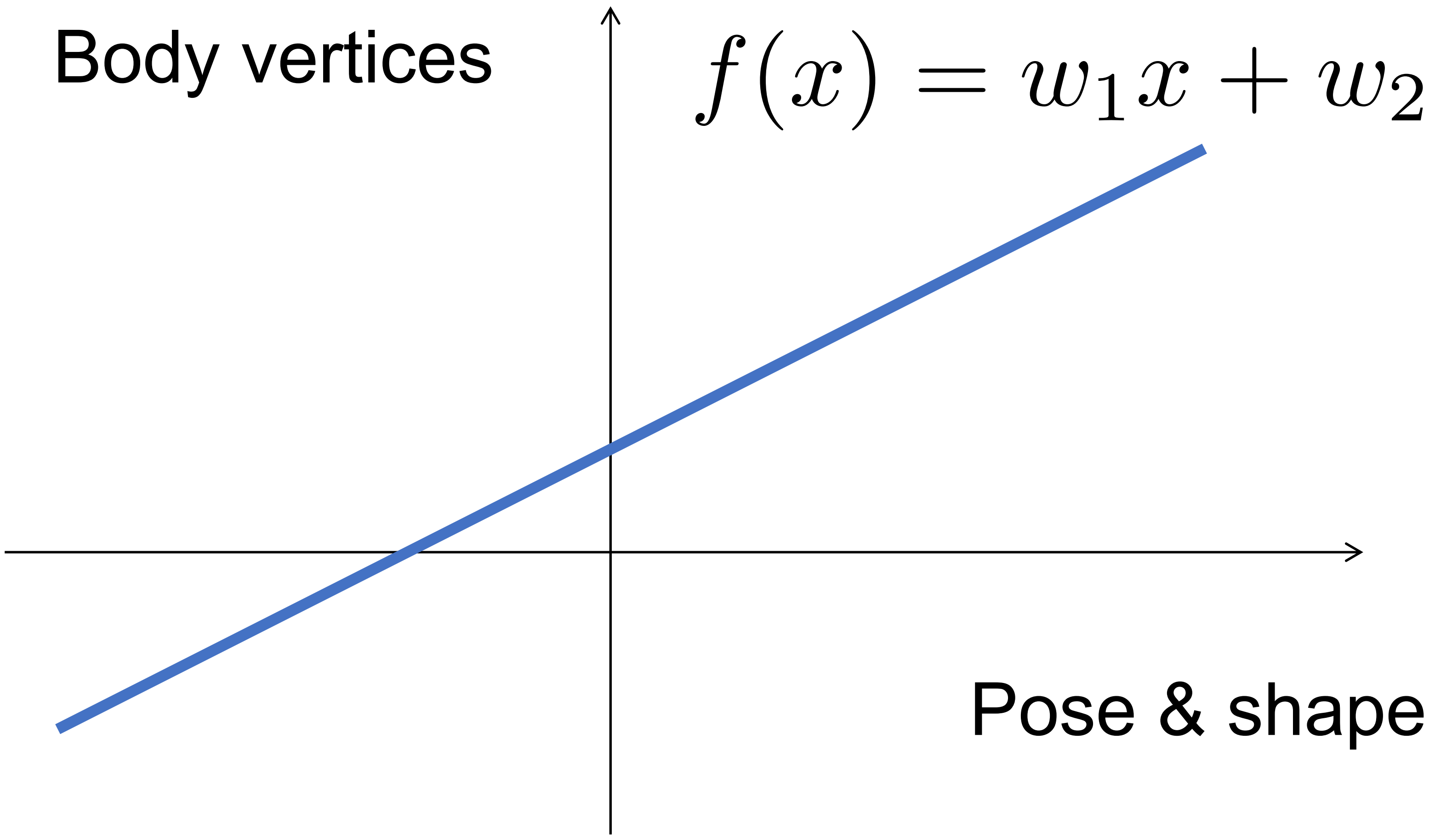


$$\mathbf{X} = \{\mathbf{X}_{\text{pose}}, \mathbf{X}_{\text{shape}}\}$$

Why should the **input X** be shape and pose ?

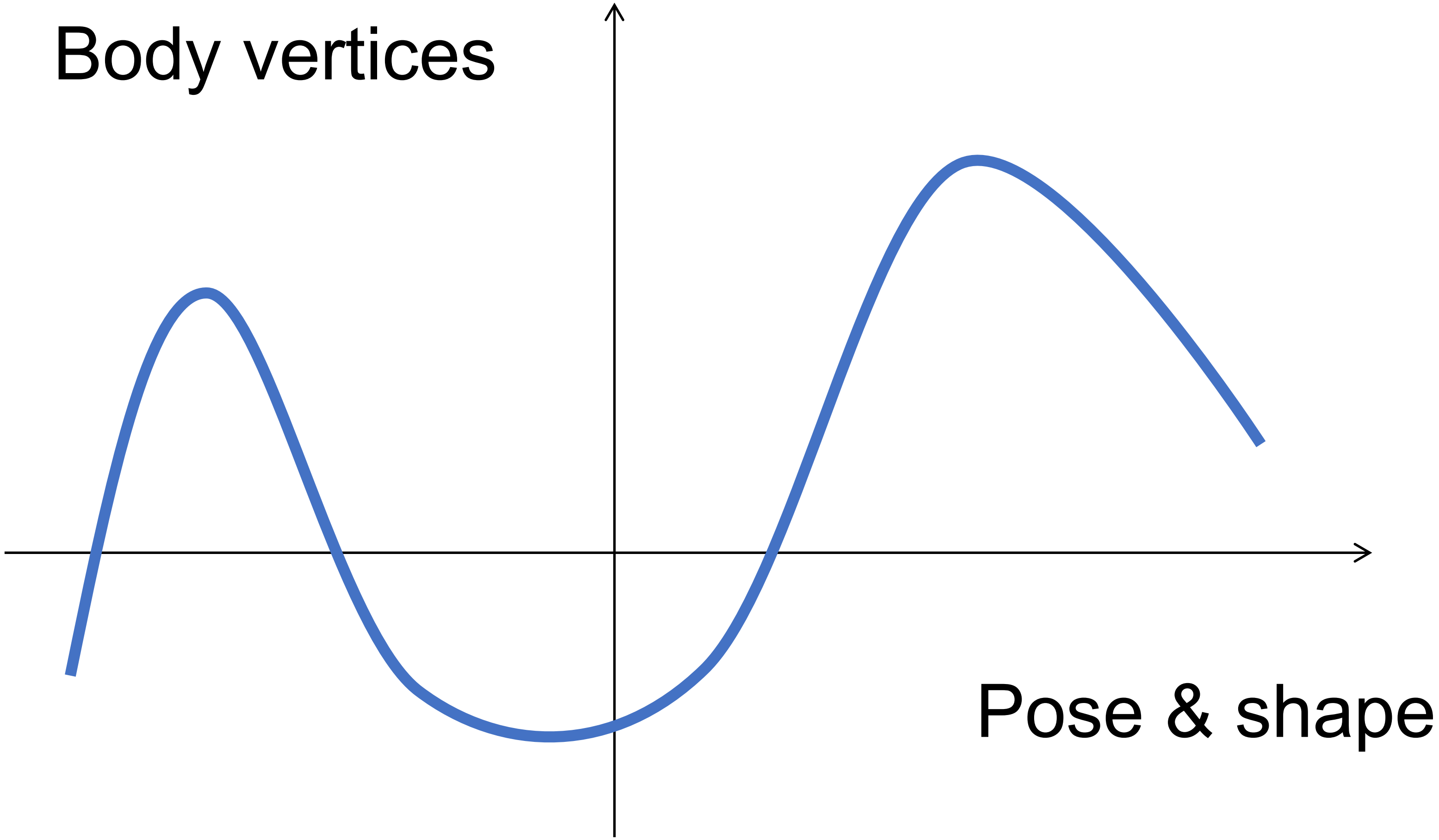
Notation: $\mathbf{X}_{\text{pose}} = \vec{\theta}$ $\mathbf{X}_{\text{shape}} = \vec{\beta}$

What kind of function ?



Linear ?

What kind of function ?



Polynomial ?

Given the function, what **w** ?

$$f(x;\mathbf{w}) = w_1 x^3 + w_2 x^2 + w_1 x + w_0$$

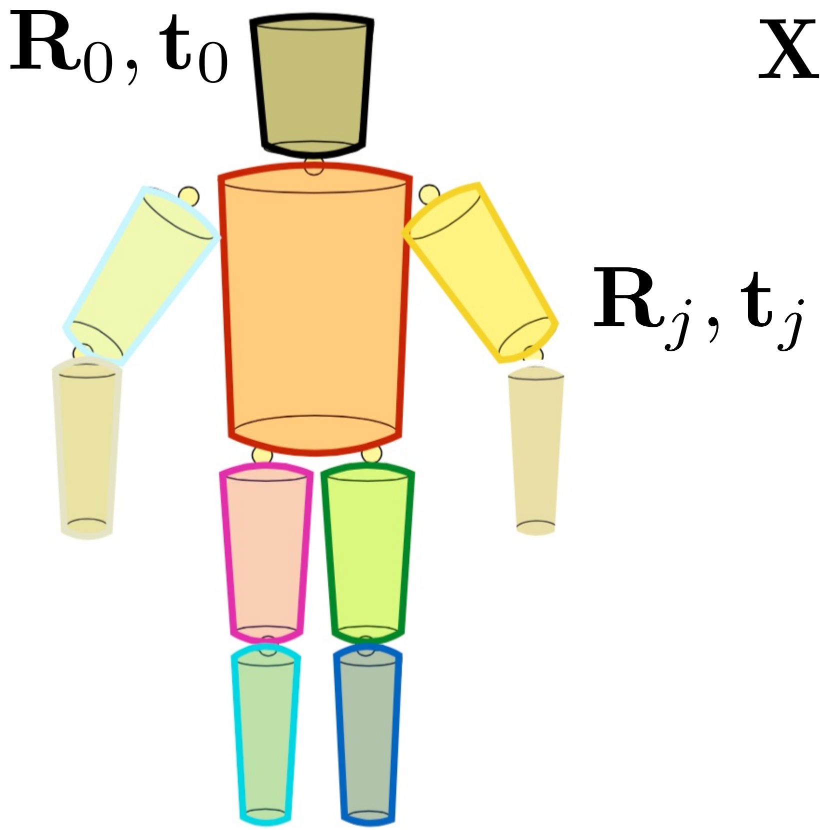
$$f(x;\mathbf{w})$$

Input parameters

Hyper-parameters

How do we parameterize pose ?

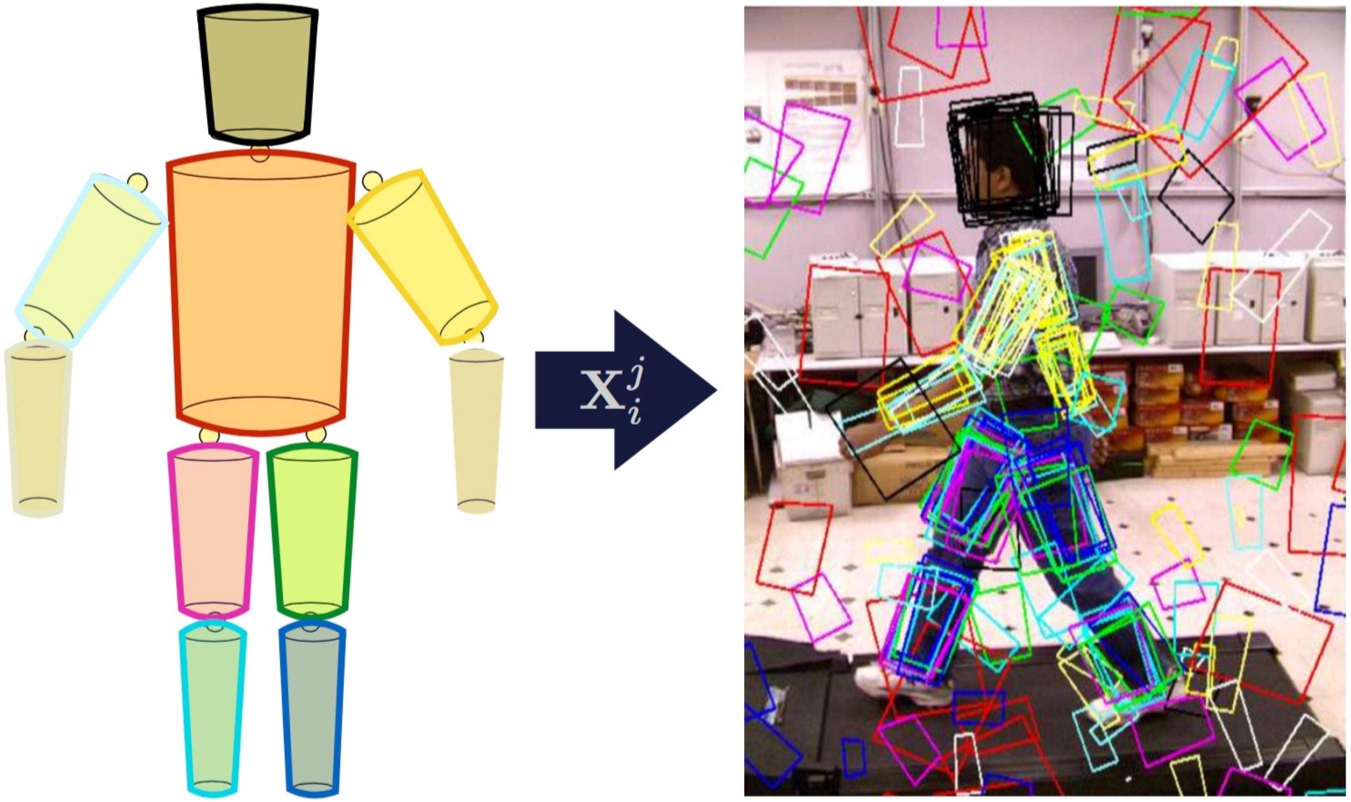
Parameterize every body part separately ?



$$\mathbf{X}_{\text{pose}} = \{\mathbf{R}_0, \mathbf{t}_0, \ldots \mathbf{R}_N, \mathbf{t}_N\}$$

Problems ?

How do we parameterize pose?

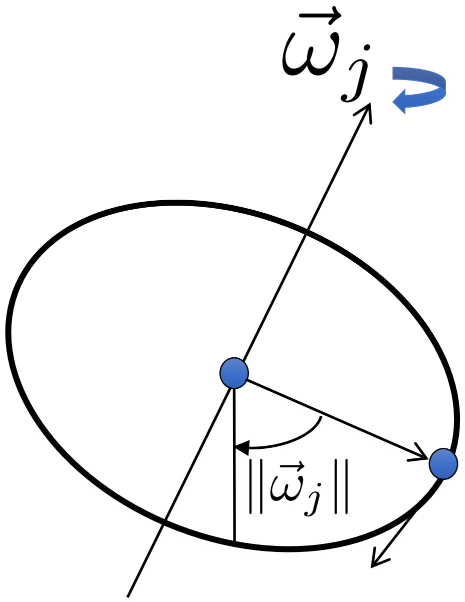


Articulated constraints not satisfied!

Rotation with Exponential Maps

$||\vec{\omega}_j||$: Angle of rotation

$\boxed{\vec{\omega}_j \text{ : scaled axis of rotation}}$



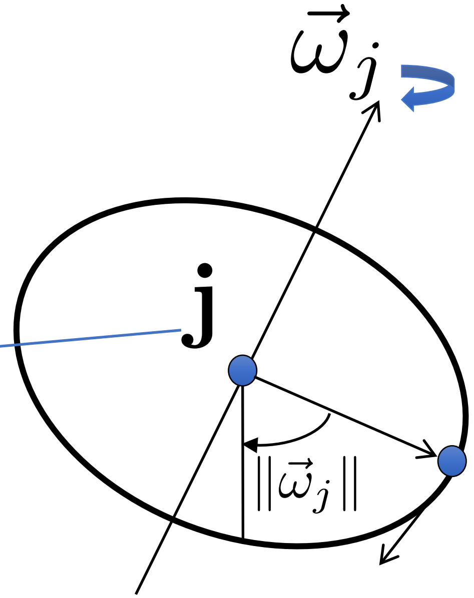
Rotation obtained with Rodrigues formula:

$$\mathbf{R} = e^{\widehat{\vec{\omega}}} = \mathcal{I} + \widehat{\omega}_j \sin(||\vec{\omega}_j||) + \widehat{\omega}^2 (1 - \cos(||\vec{\omega}_j||)$$

Joint Rigid Body Motion

The transformation associated with a rotational joint is:

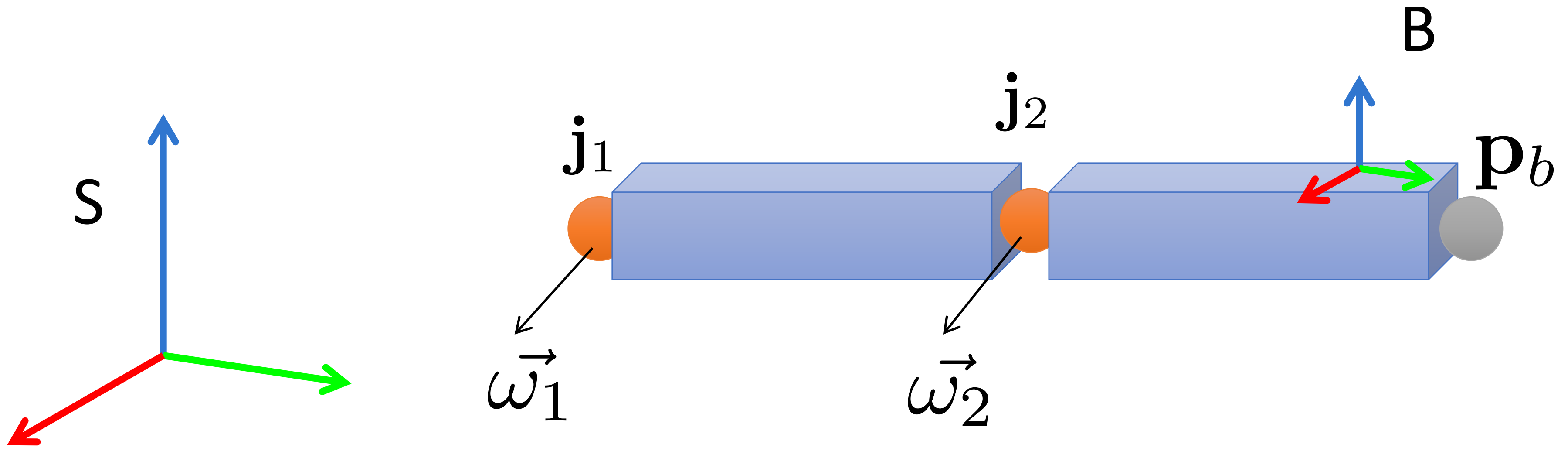
Joint location



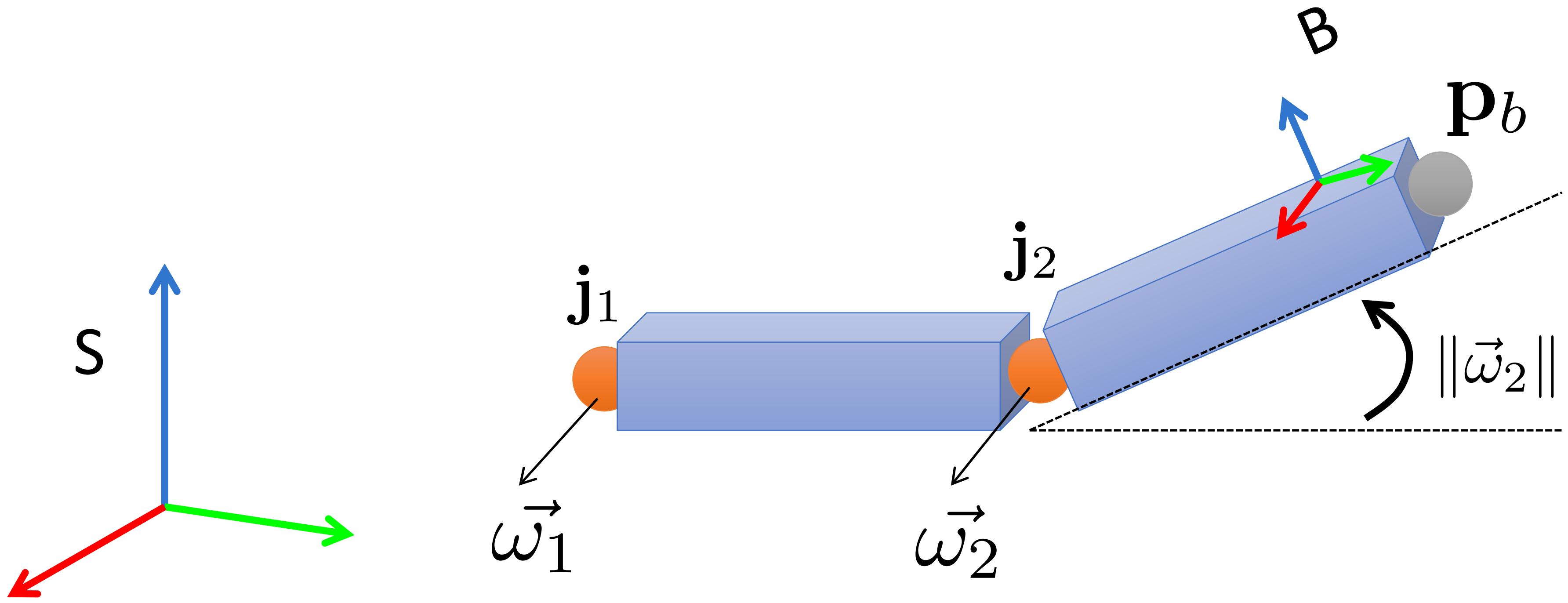
$$G(\vec{\omega}) = \begin{bmatrix} [e^{\vec{\omega}}]_{3\times 3} & \mathbf{j}_{3\times 1} - \mathbf{j}_{\text{parent}} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix}$$

Rigid Body Motion

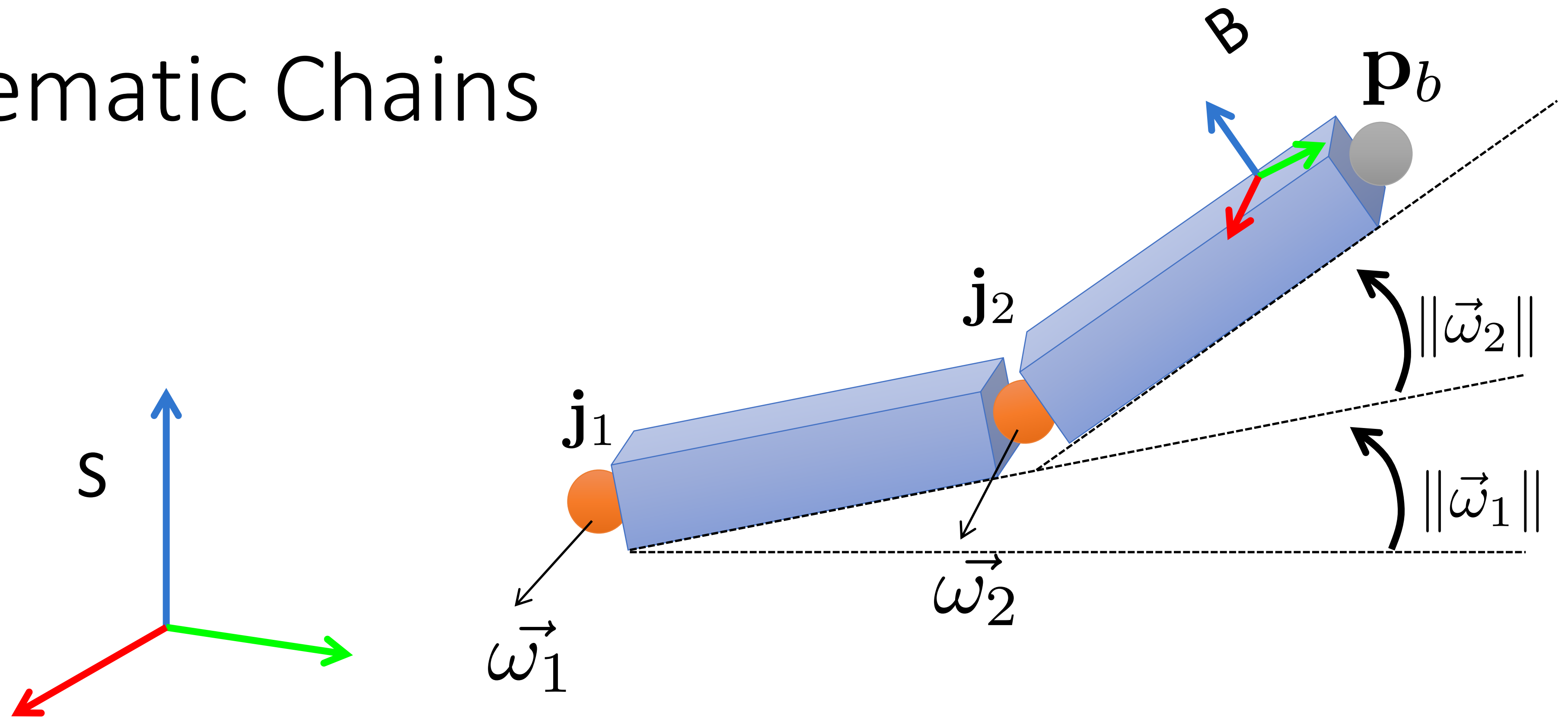
Kinematic Chains



Kinematic Chains



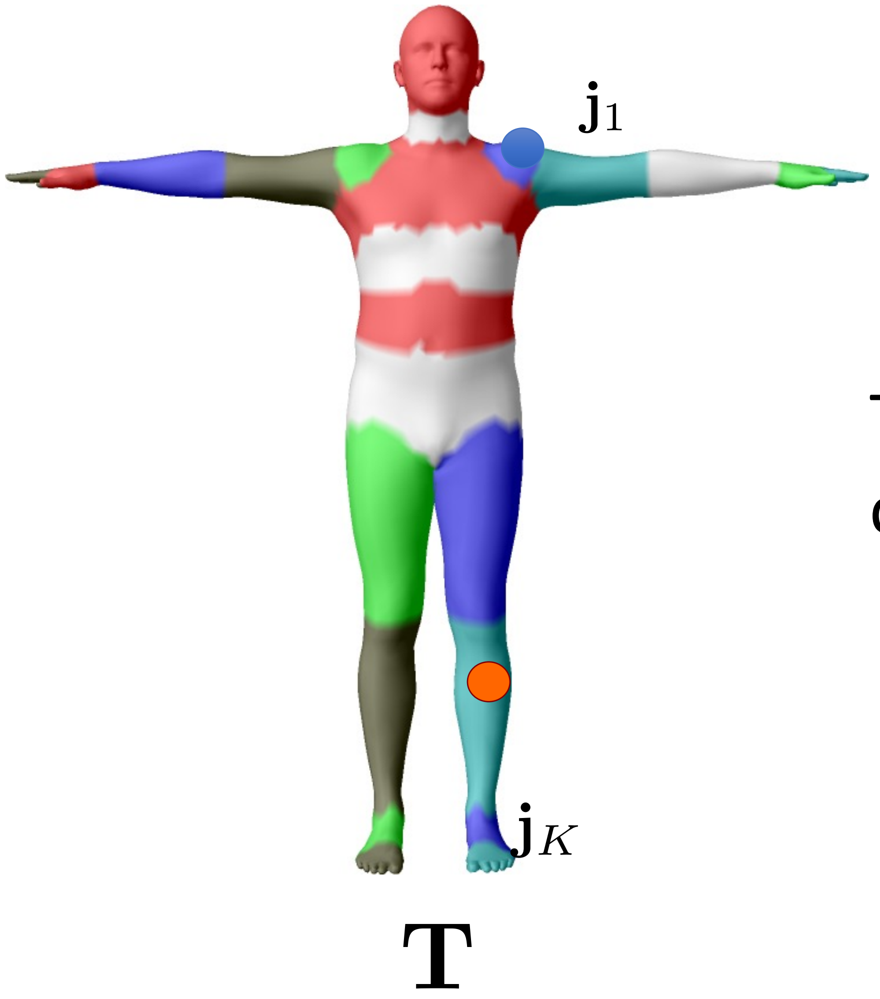
Kinematic Chains



The coordinates of the point in the spatial frame are:

$$\bar{\mathbf{p}}_s = G(\vec{\omega_1}, \vec{\omega_2}, \mathbf{j}_1, \mathbf{j}_2) = G(\vec{\omega}_1, \mathbf{j}_1)\boxed{G(\vec{\omega_2}, \mathbf{j}_2)}\bar{\mathbf{p}}_b$$

Pose Parameters



Given a set of joint locations

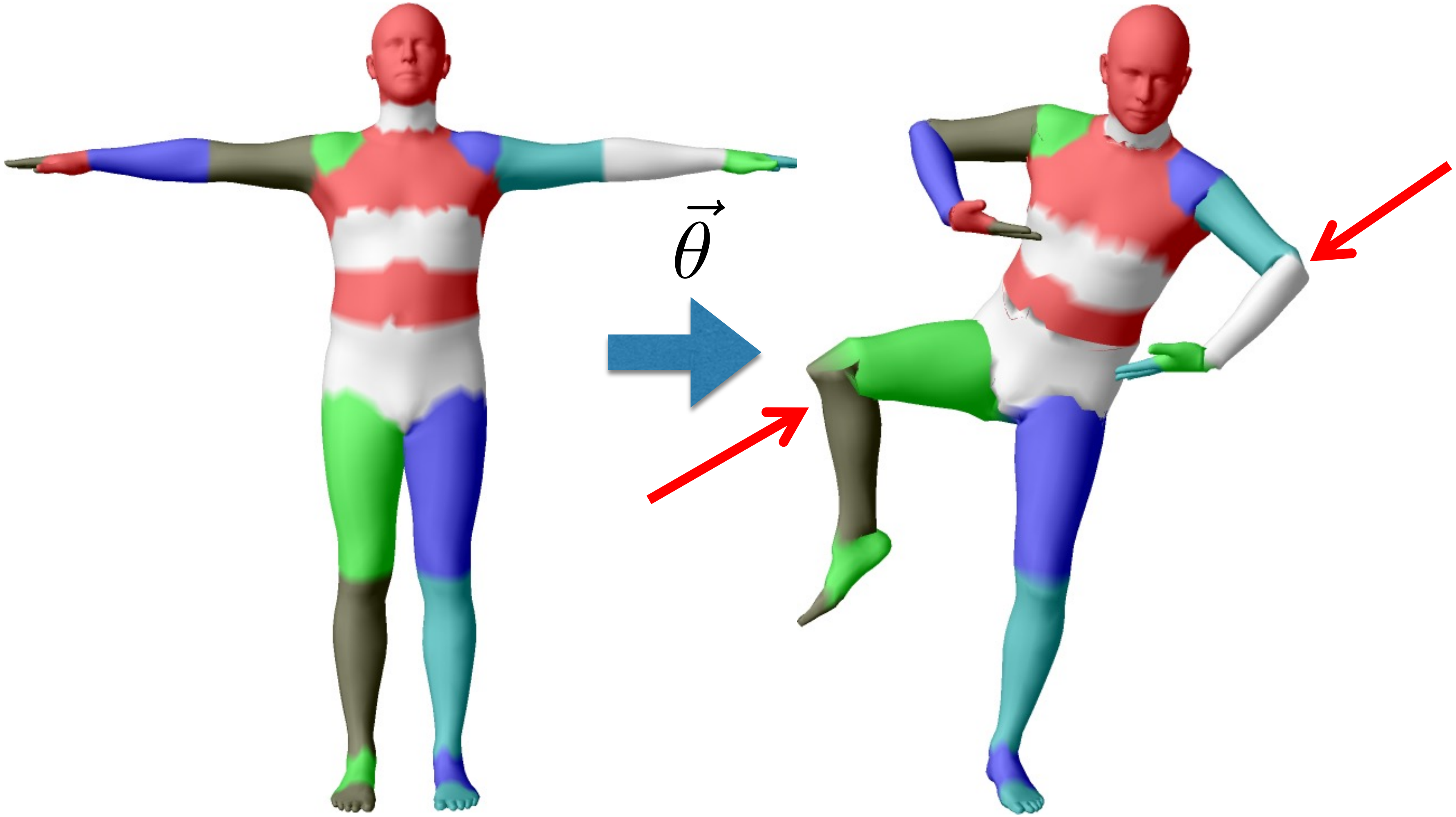
$$\mathbf{J} = (\mathbf{j}_1, \ldots, \mathbf{j}_K)^T$$

The pose defined as the vector of concatenated part axis-angles

$$\vec{\theta} = (\vec{\omega}_1, \ldots, \vec{\omega}_k)^T$$

Pons-Moll & Rosenhahn 2011
Model-based Pose Estimation. Looking at People.

Kinematic Chain Problems

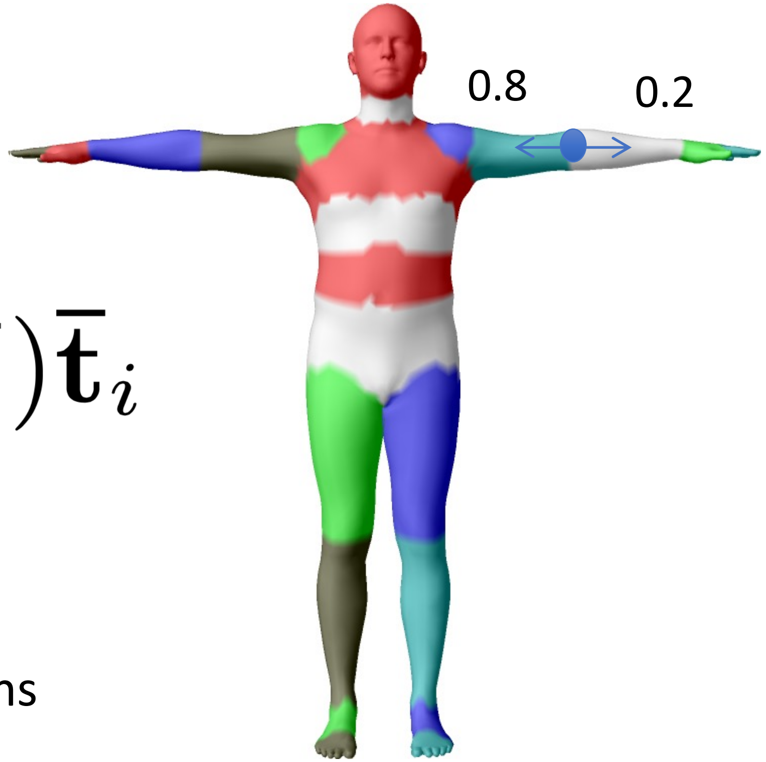


Linear Blend Skinning

$$\bar{\mathbf{t}}'_i = \sum_{k=1}^{K} w_{k,i} G'_k(\vec{\theta}, \mathbf{J}) \bar{\mathbf{t}}_i$$

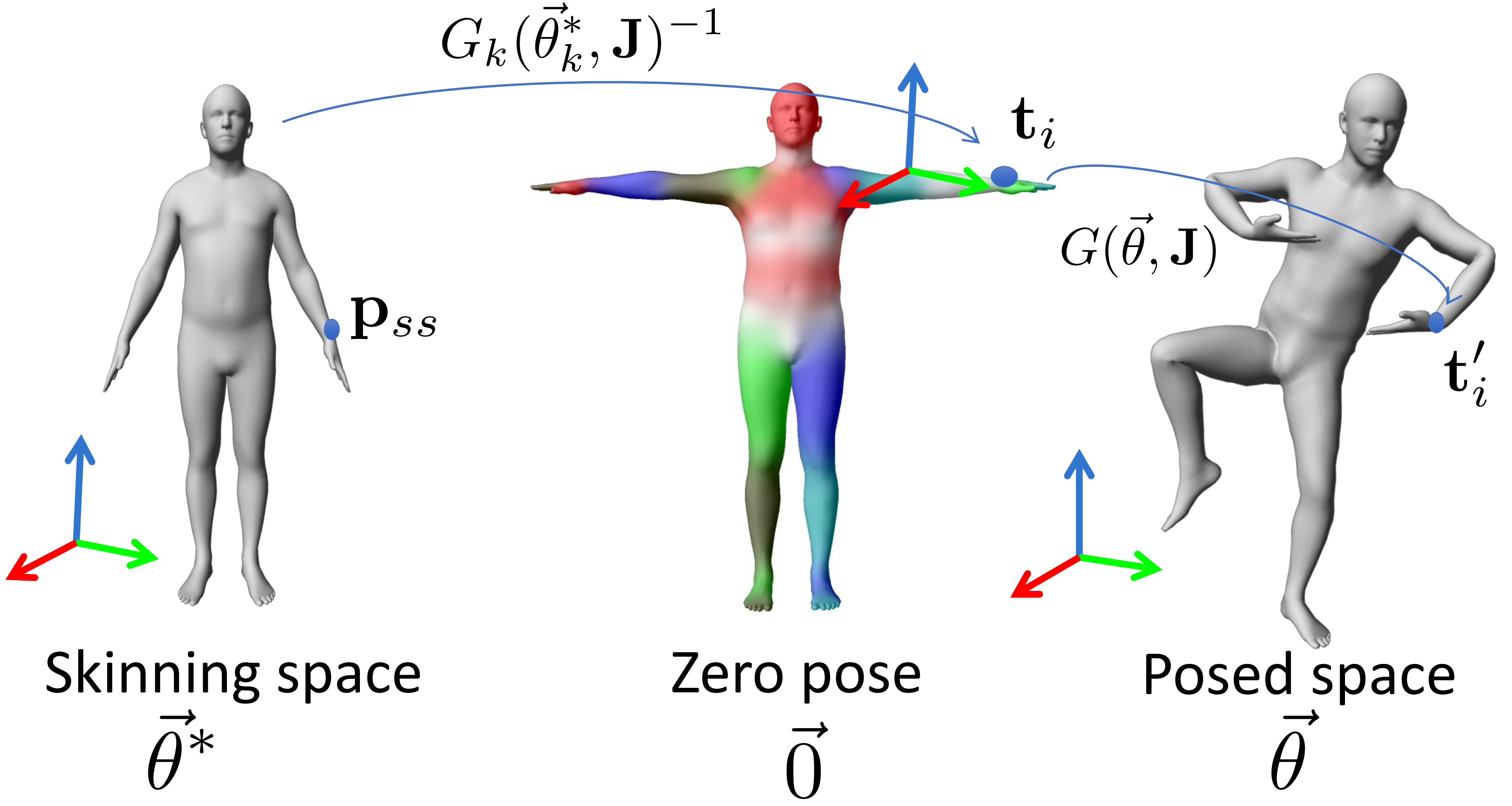
Blend weights

Part transformations



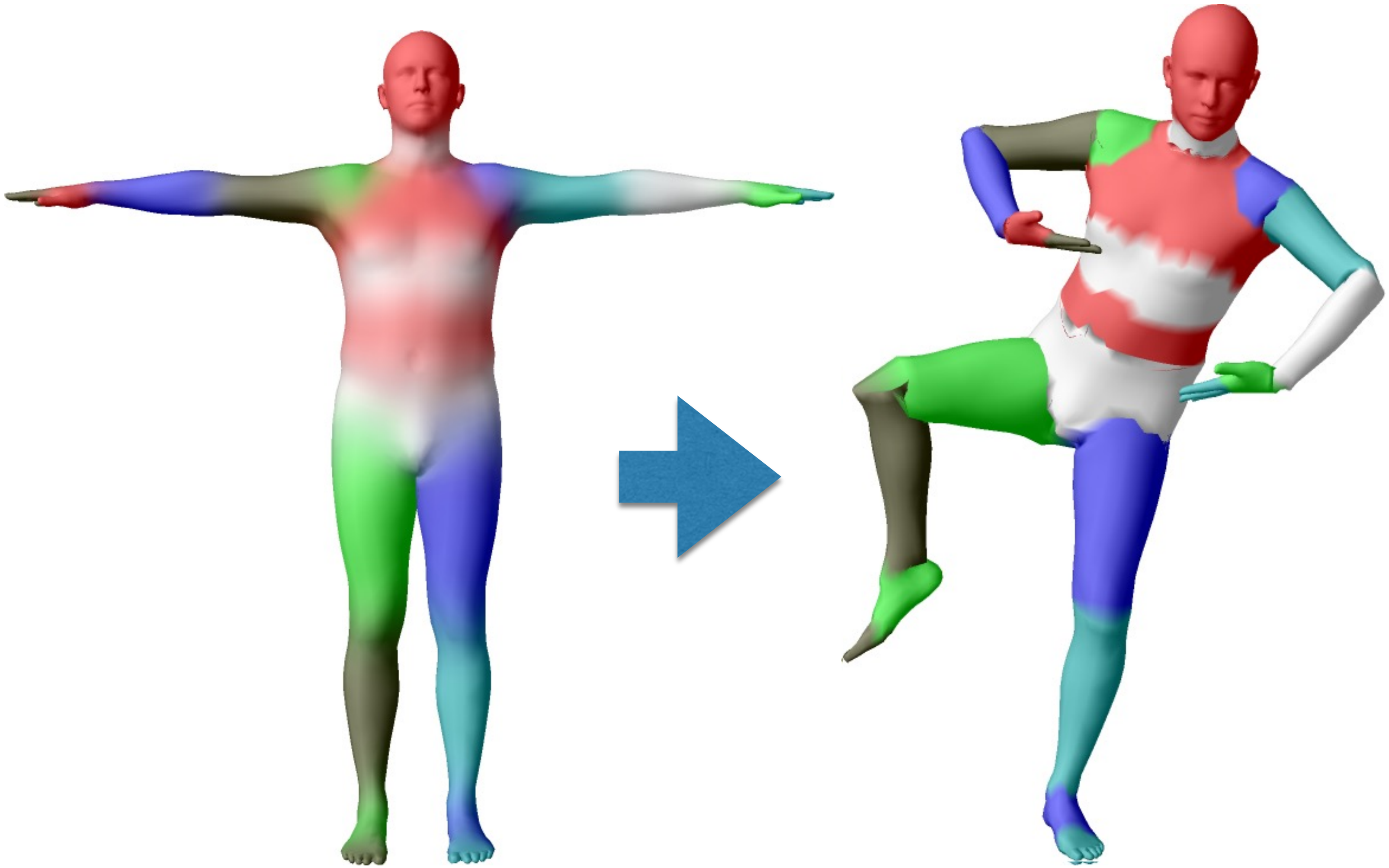
Points transformed as blended linear combination of joint transformation matrices

Binding Matrices



$$\boxed{\bar{\mathbf{t}}_i = G_k(\vec{\theta}_k, \mathbf{J})^{-1}\mathbf{p}_{ss}} \Rightarrow G'_k(\vec{\theta}, \mathbf{J}) = G_k(\vec{\theta}, \mathbf{J})\boxed{G_k(\vec{\theta}^*, \mathbf{J})^{-1}}$$

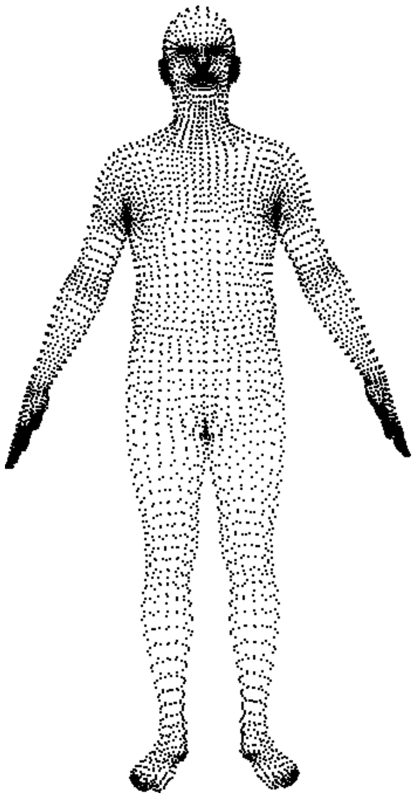
Linear Blend Skinning



Standard Skinning

Standard skinning produces vertices from…

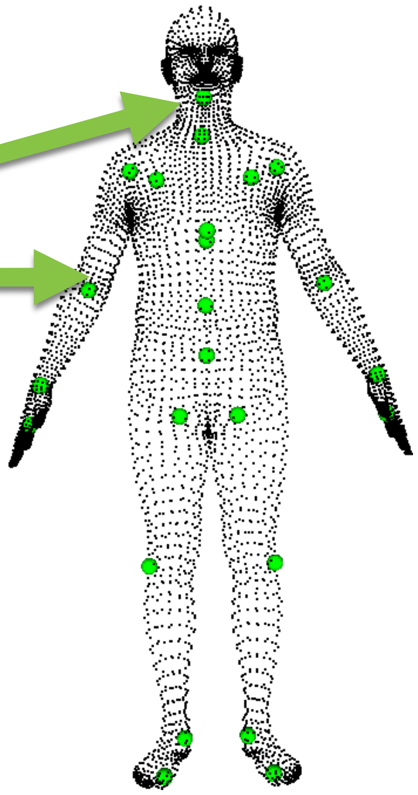
- Rest pose vertices: $\mathbf{T} \in \mathbb{R}^{3N}$
- Joint locations: $\mathbf{J} \in \mathbb{R}^{3K}$
- Weights: $\mathcal{W} \in \mathbb{R}^{N \times K}$
- Pose parameters: $\vec{\theta} \in \mathbb{R}^{3K}$



Standard Skinning

Standard skinning produces vertices from...

- Rest pose vertices: $\mathbf{T} \in \mathbb{R}^{3N}$
- $\boxed{\text{Joint locations: } \mathbf{J} \in \mathbb{R}^{3K}}$
- Weights: $\mathcal{W} \in \mathbb{R}^{N \times K}$
- Pose parameters: $\vec{\theta} \in \mathbb{R}^{3K}$



Standard Skinning

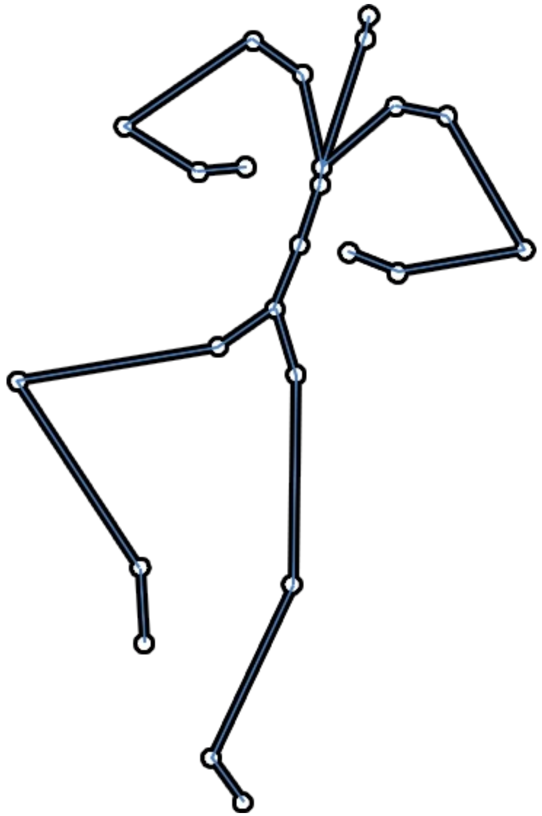
Standard skinning produces vertices from…

- Rest pose vertices: $\mathbf{T} \in \mathbb{R}^{3N}$
- Joint locations: $\mathbf{J} \in \mathbb{R}^{3K}$
- Weights: $\mathcal{W} \in \mathbb{R}^{N \times K}$
- Pose parameters: $\vec{\theta} \in \mathbb{R}^{3K}$

Standard Skinning

Standard skinning produces vertices from…

- Rest pose vertices: $\mathbf{T} \in \mathbb{R}^{3N}$
- Joint locations: $\mathbf{J} \in \mathbb{R}^{3K}$
- Weights: $\mathcal{W} \in \mathbb{R}^{N \times K}$
- Pose parameters: $\vec{\theta} \in \mathbb{R}^{3K}$

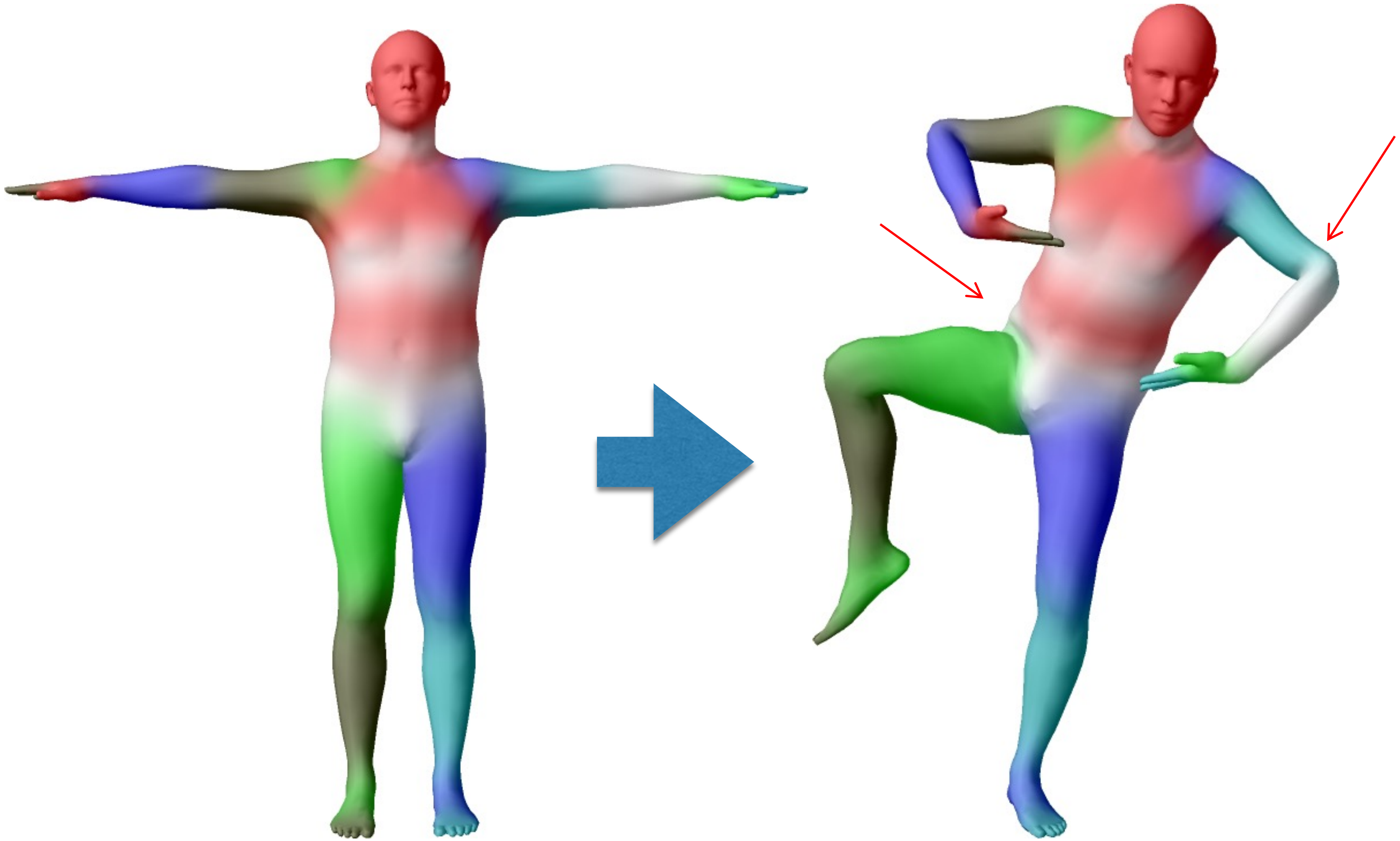


Skinning function

- Rest pose vertices: $\mathbf{T} \in \mathbb{R}^{3N}$
- Joint locations: $\mathbf{J} \in \mathbb{R}^{3K}$
- Weights: $\mathcal{W} \in \mathbb{R}^{N \times K}$
- Pose parameters: $\vec{\theta} \in \mathbb{R}^{3K}$

$$\boxed{W(\mathbf{T}, \mathbf{J}, \mathcal{W}, \vec{\theta}) \mapsto \text{vertices}}$$

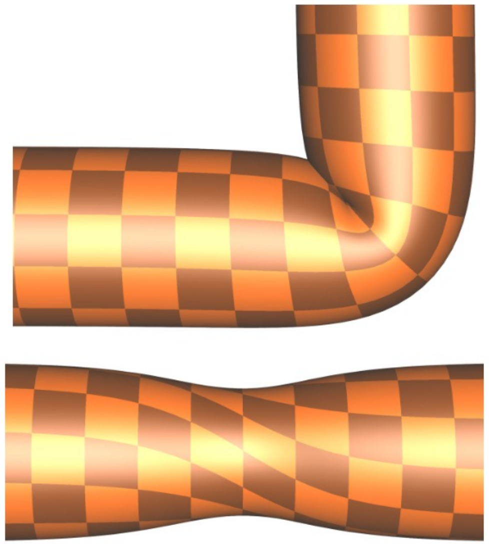
LBS problems



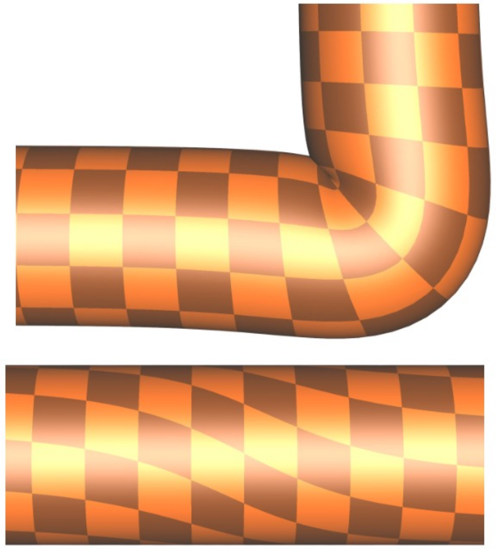
Standard Skinning

Problem: We want better pose-driven changes

LBS Skinning: collapse



DQ Skinning: bulge

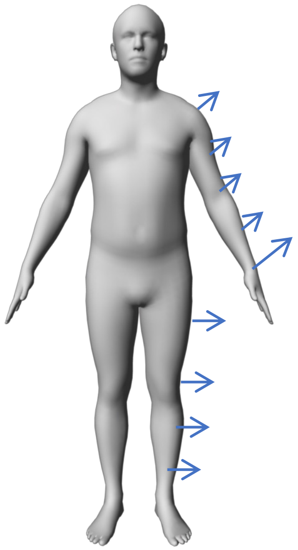


[Kavan et al., 2012]

Solution: Blend Shapes

A **blend shape** is a set of vertex displacements in a rest pose

- Pose blend shapes: correct for LBS problems

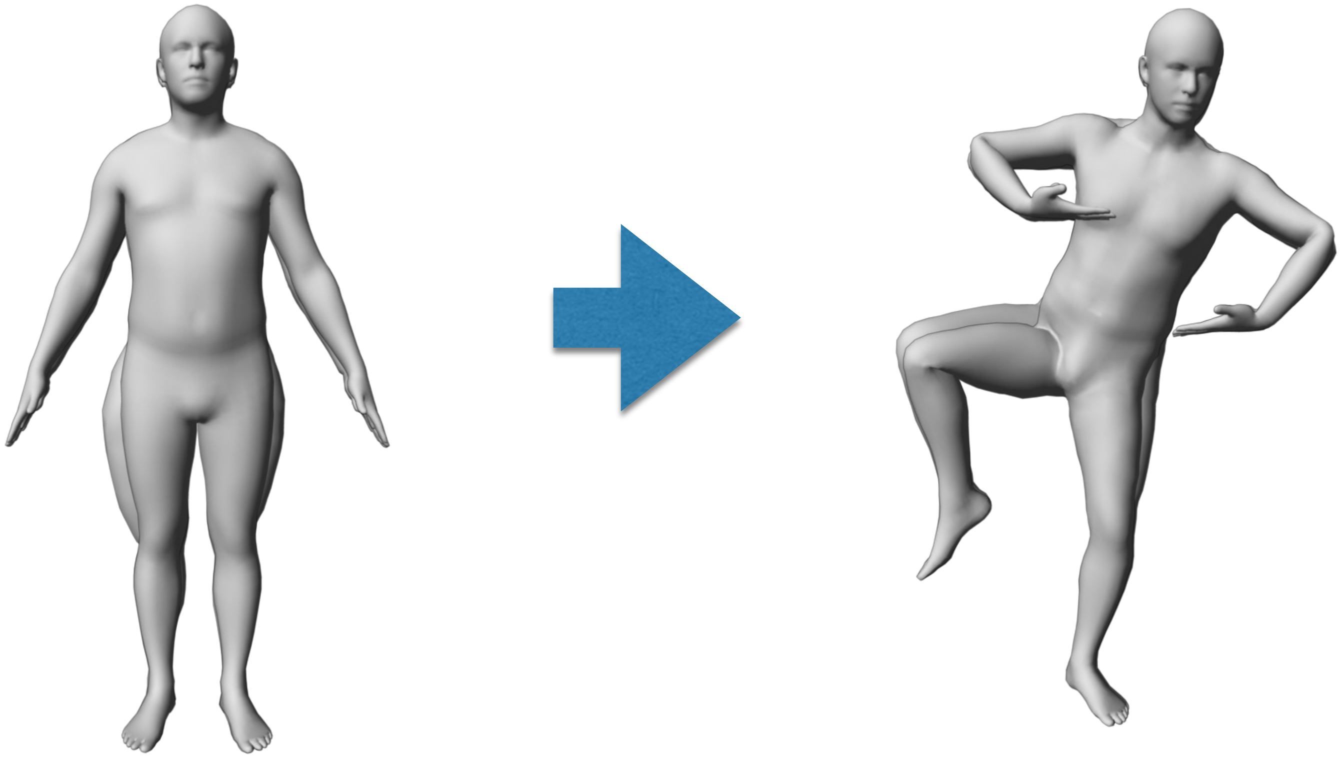


$$\mathbf{P} = \mathrm{vec}\left(\begin{bmatrix} \Delta x_1 & \Delta y_1 & \Delta z_1 \\ & \vdots & \\ & \vdots & \\ \Delta x_N & \Delta y_N & \Delta z_N \end{bmatrix}\right) \in \mathbb{R}^{3N}$$

$\longrightarrow$ Offset 1

Pose Blend Shapes

With blend shape correction



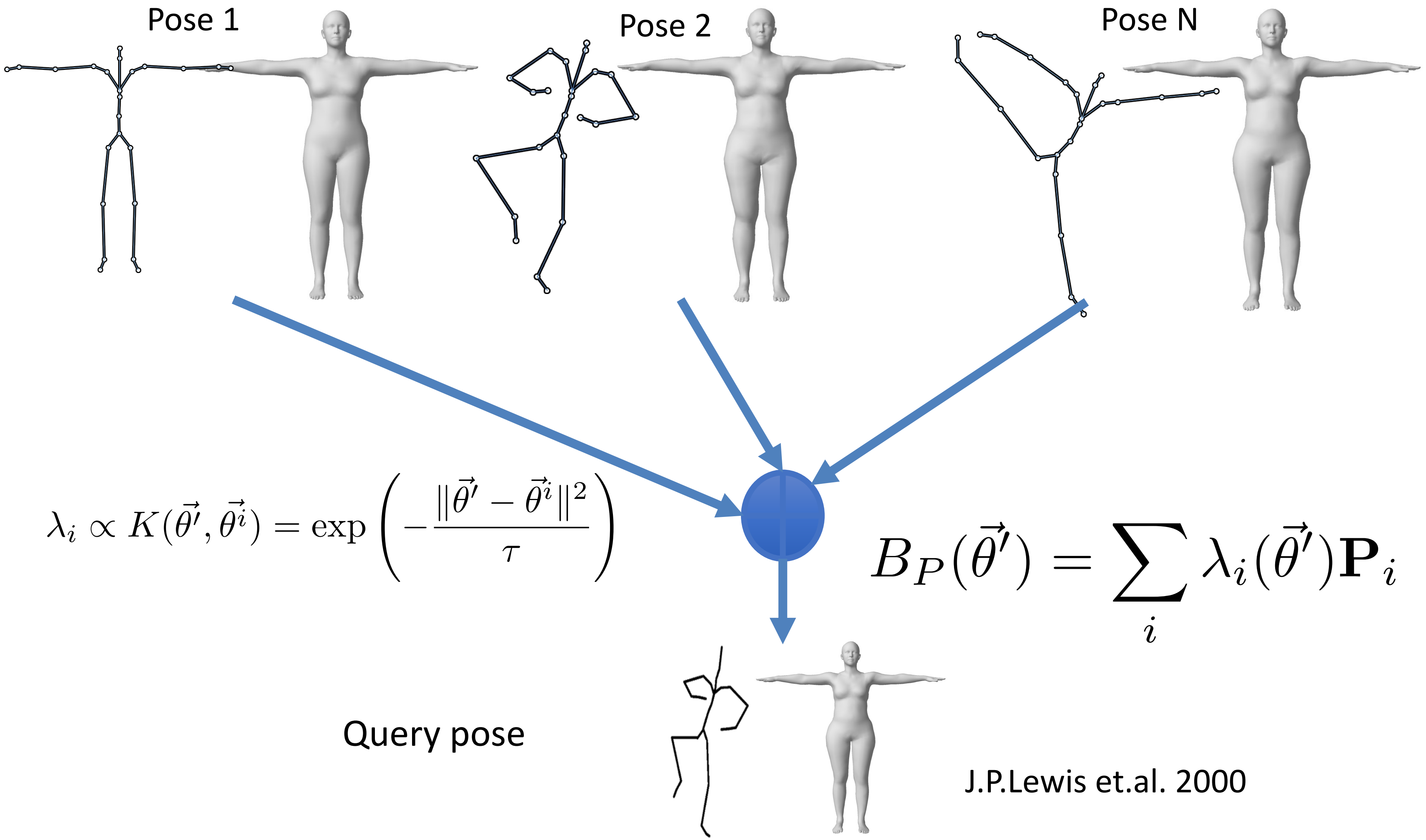
How to predict Blend Shapes ?

- Animators sculpt it manually!

- Time consuming, does not scale

Can we leverage training data ?

Scattered Data Interpolation



$$\lambda_i \propto K(\vec{\theta}', \vec{\theta}^i) = \exp\left(-\frac{\|\vec{\theta}' - \vec{\theta}^i\|^2}{\tau}\right)$$

$$B_P(\vec{\theta}') = \sum_i \lambda_i(\vec{\theta}')\mathbf{P}_i$$

J.P.Lewis et.al. 2000

Problems Scattered Data Interpolation

- Computationally expensive (need to find closest poses in a database)
- Does not extrapolate very well to novel poses

Problems

- If we don't use scattered data interpolation, how do we define pose blend shapes ?

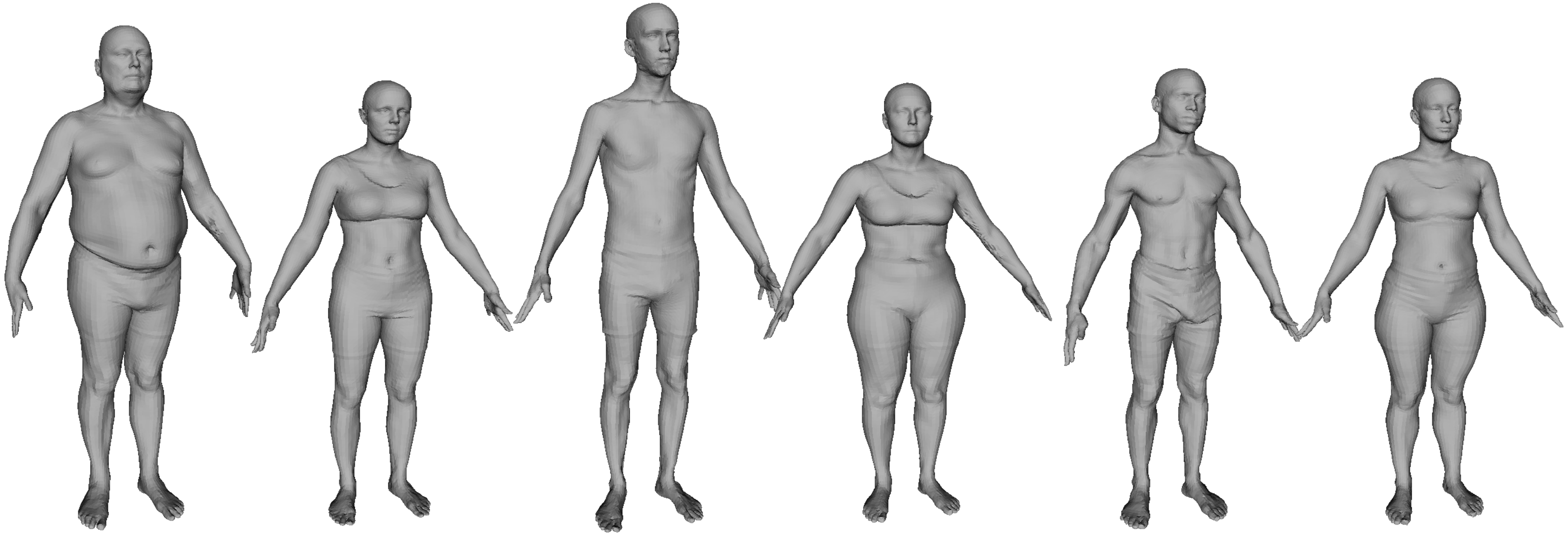
$$B_P(\vec{\theta}')$$

- How to set the skinning parameters ?

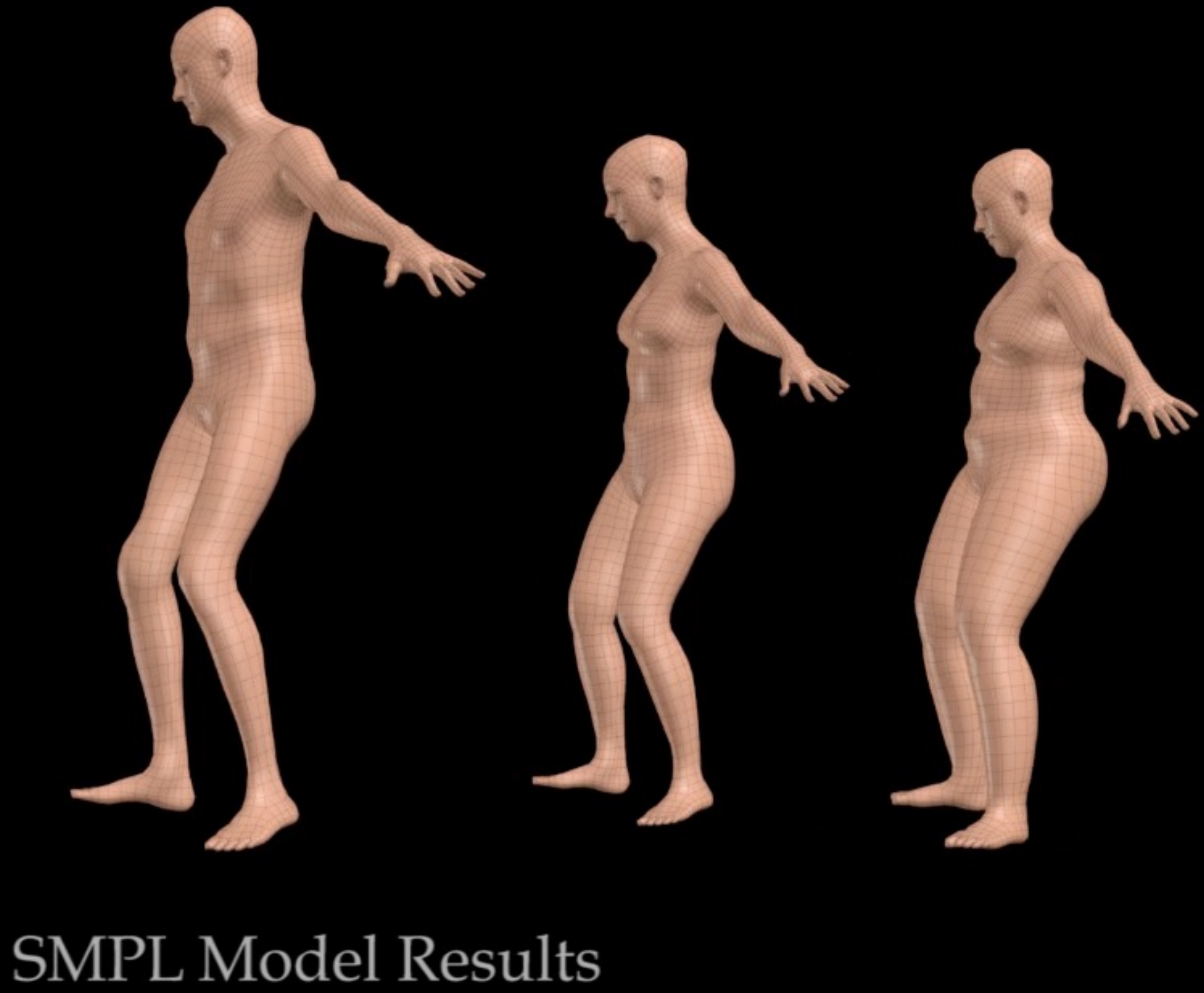
$$\mathbf{T} \in \mathbb{R}^{3N} \quad \mathbf{J} \in \mathbb{R}^{3K} \quad \mathcal{W} \in \mathbb{R}^{N \times K}$$

More Problems

How do we model shape identity variations ?



SMPL



SMPL Philosophy

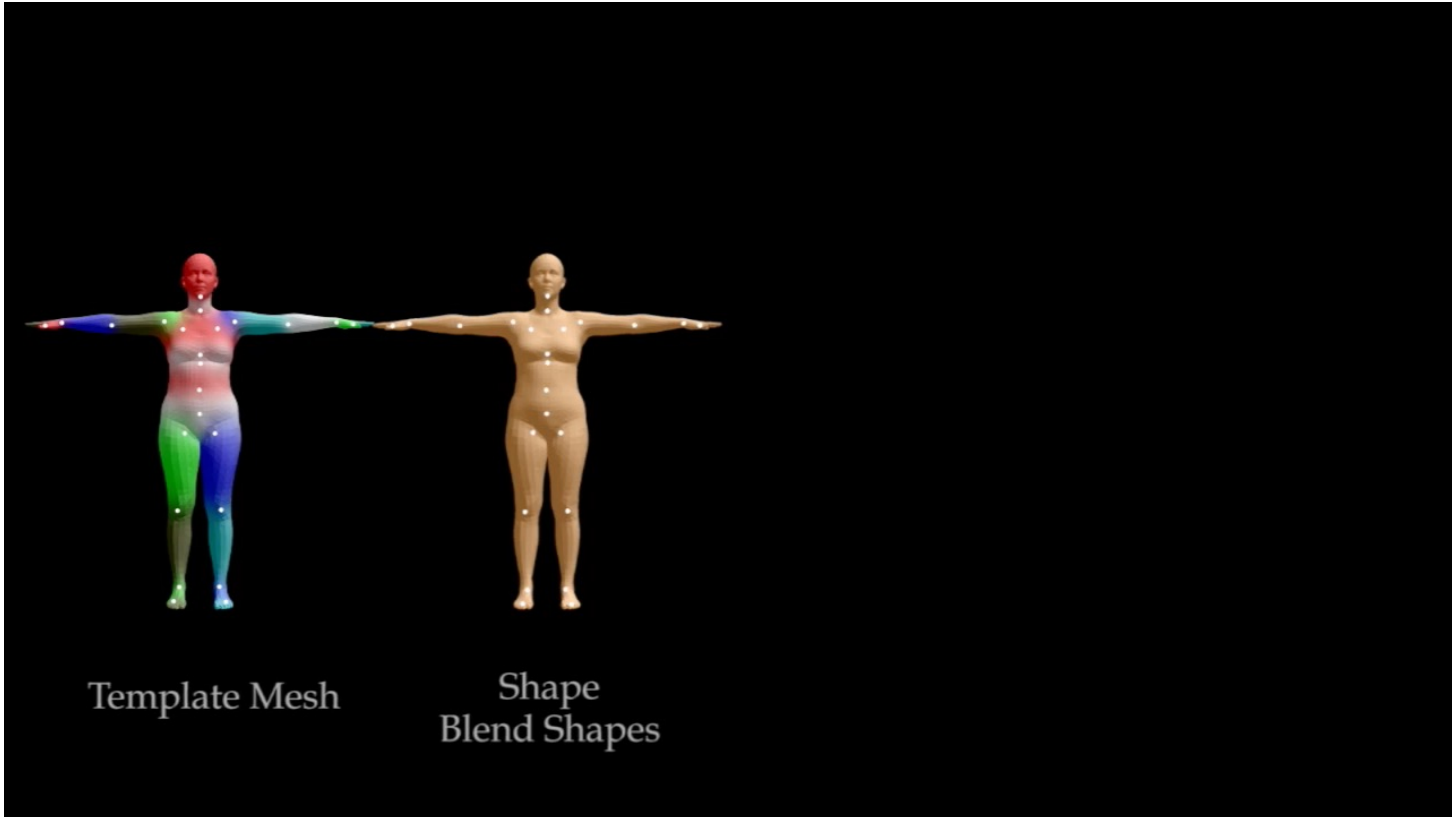
We aim for the simplest possible model while having state-of-the-art performance

- Makes training easier
- Enables compatibility

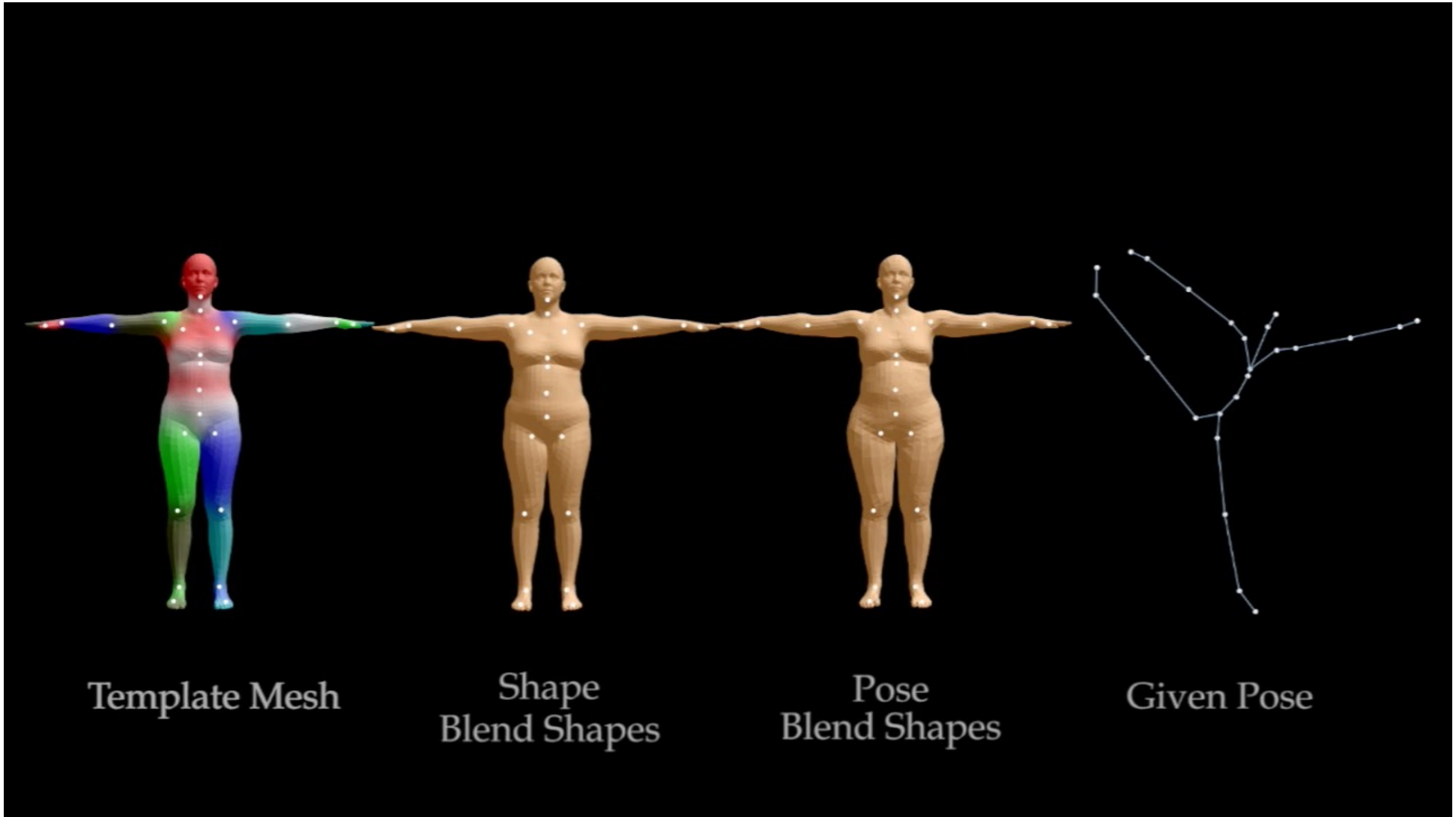
Loper et.al. SIGGRAPH Asia, 2015

Pipeline

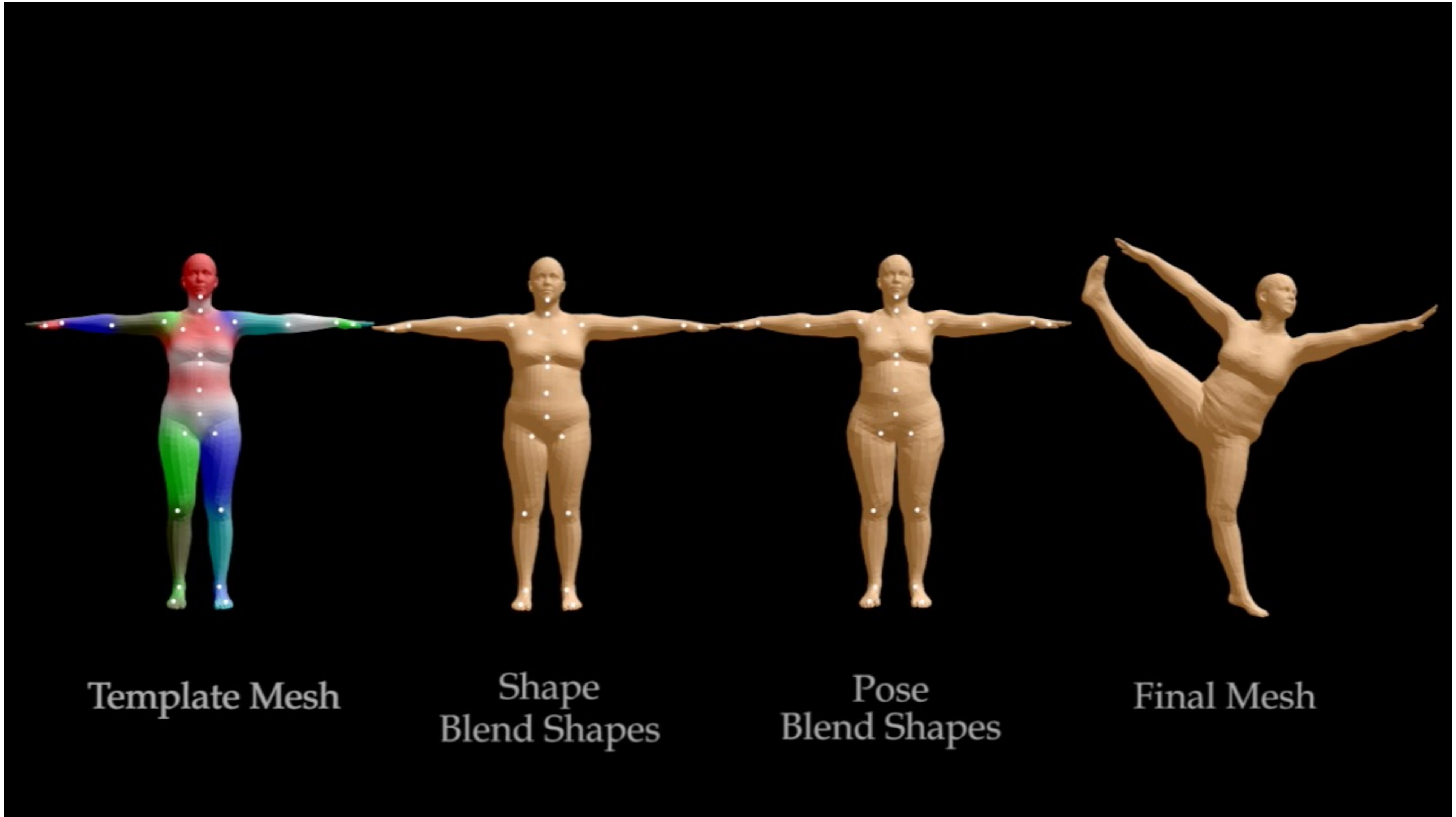
Pipeline



Pipeline


Template Mesh
Shape
Blend Shapes
Pose
Blend Shapes
Given Pose

Pipeline


Template Mesh
Shape
Blend Shapes
Pose
Blend Shapes
Final Mesh

Parameterized Skinning

Standard skinning $W(\mathbf{T}, \mathbf{J}, \mathcal{W}, \vec{\theta}) \mapsto \text{vertices}$

SMPL model

$$M(\vec{\theta}, \vec{\beta}) = W(\mathbf{T}_F(\vec{\beta}, \theta), \mathbf{J}(\vec{\beta}), \mathcal{W}, \vec{\theta}) \mapsto \text{vertices}$$

SMPL is skinning parameterized by pose $\vec{\theta}$ and shape $\vec{\beta}$

SMPL: BS are a parametric function of pose

We parameterize the skinning equation by pose

$$W(\mathbf{T}, \mathbf{J}, \mathcal{W}, \vec{\theta})$$

$$\Downarrow$$

$$W(T(\theta), \mathbf{J}, \mathcal{W}, \vec{\theta})$$

Parameterized Skinning

$$W\left(T(\theta), \mathbf{J}, \mathcal{W}, \vec{\theta}\right) \mapsto \text{vertices}$$

$$T(\vec{\theta}) = \mathbf{T} + B_P(\vec{\theta})$$

Our rest vertices are linear in $f(\theta)$

$$B_P(\vec{\theta}) = \sum_i^{|f(\vec{\theta})|} f_i(\vec{\theta})\mathbf{P}_i$$

Each is
a blend shape

Parameterized Skinning

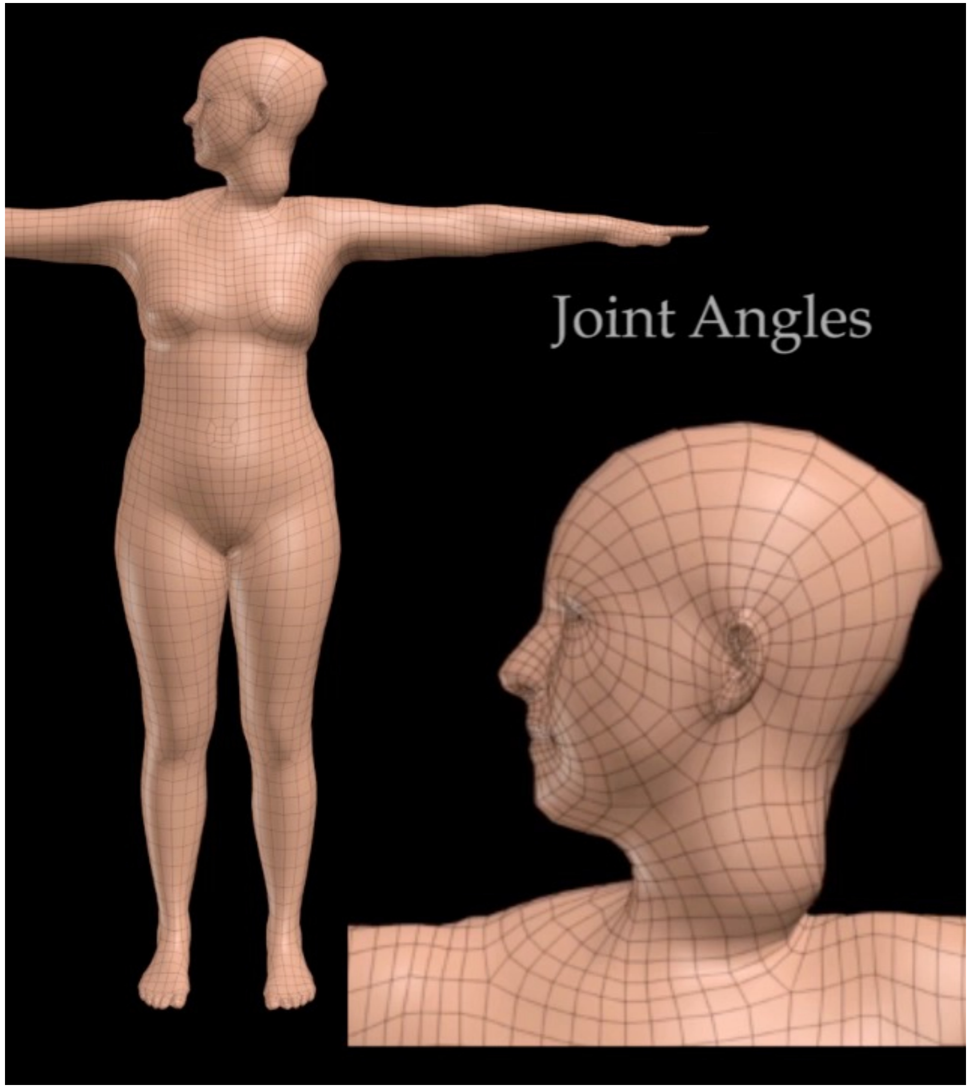
- What function $f(\vec{\theta})$?

$$B_P(\vec{\theta}) = \sum_i^{|f(\vec{\theta})|} f_i(\vec{\theta}) \mathbf{P}_i$$

- Simplest possible:

$$f(\vec{\theta}) = \vec{\theta}$$

Neck Rotation



Parameterized Skinning

- What function $f(\vec{\theta})$?

$$B_P(\vec{\theta}) = \sum_i^{|f(\vec{\theta})|} f_i(\vec{\theta})\mathbf{P}_i$$

- Idea: we consider $f(\vec{\theta})$ as the vectorized joint rotation matrices

- Blend shapes are *linear in rotation matrix elements*

Pose Blend Shapes

$$B_P(\vec{\theta}) = \sum_{i}^{|f(\vec{\theta})|} f_i(\vec{\theta})\mathbf{P}_i$$

$$\vec{\theta} = (\vec{\omega}_1, \ldots, \vec{\omega}_k)^T$$

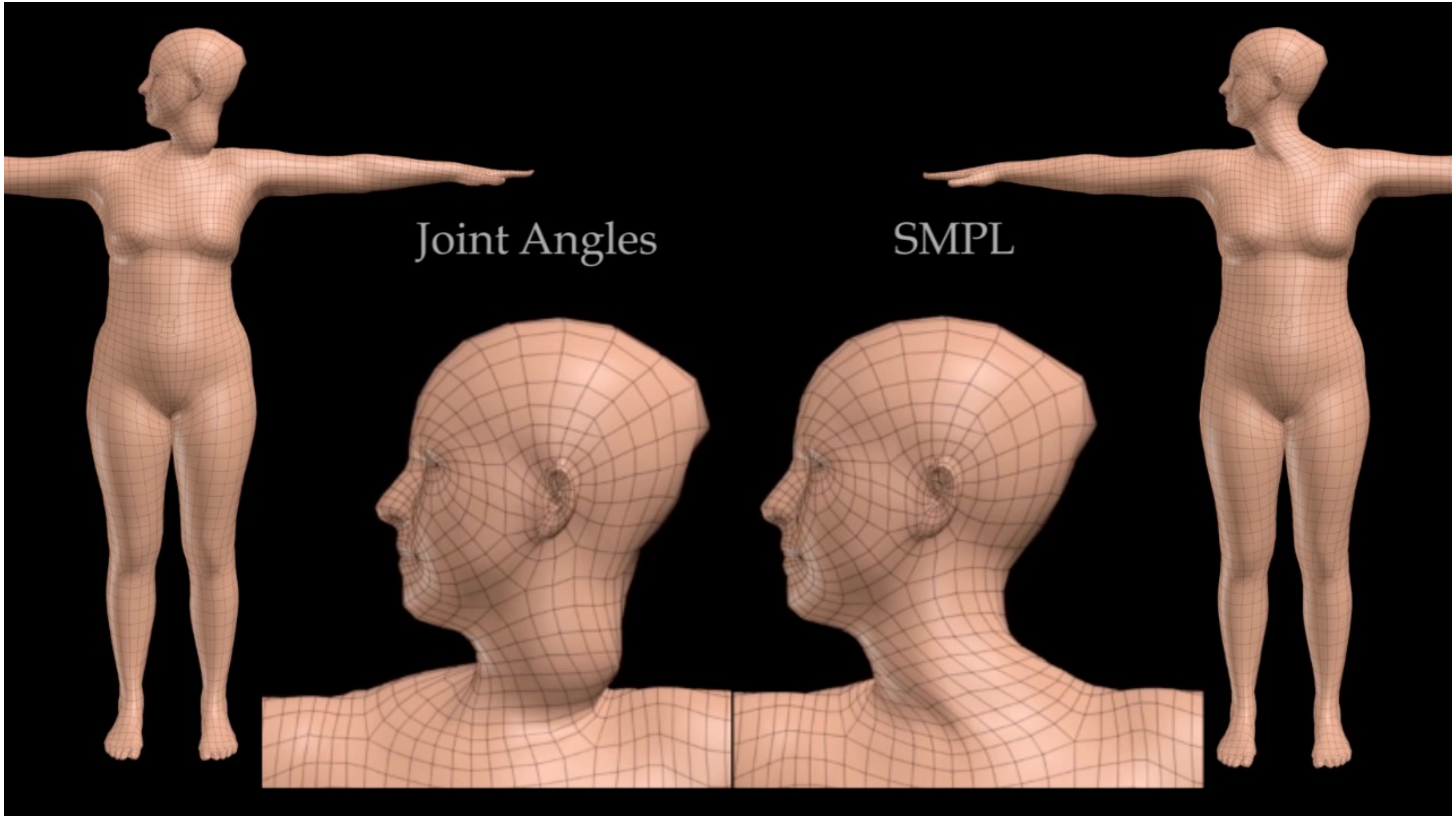
$$e^{\hat{\omega}_1} - \mathcal{I} \qquad e^{\hat{\omega}_K} - \mathcal{I}$$

Not a minus

$$f(\vec{\theta}) = [\underbrace{\bar{e}^{\hat{\omega}_1}_{1,1} \ldots \bar{e}^{\hat{\omega}_1}_{3,3}} \quad \ldots \quad \underbrace{\bar{e}^{\hat{\omega}_K}_{1,1} \ldots \bar{e}^{\hat{\omega}_K}_{3,3}}]$$

9 elements of the rotation matrix-> We learn 9xK=207 blendshapes

Neck Rotation



Joint Location Estimation

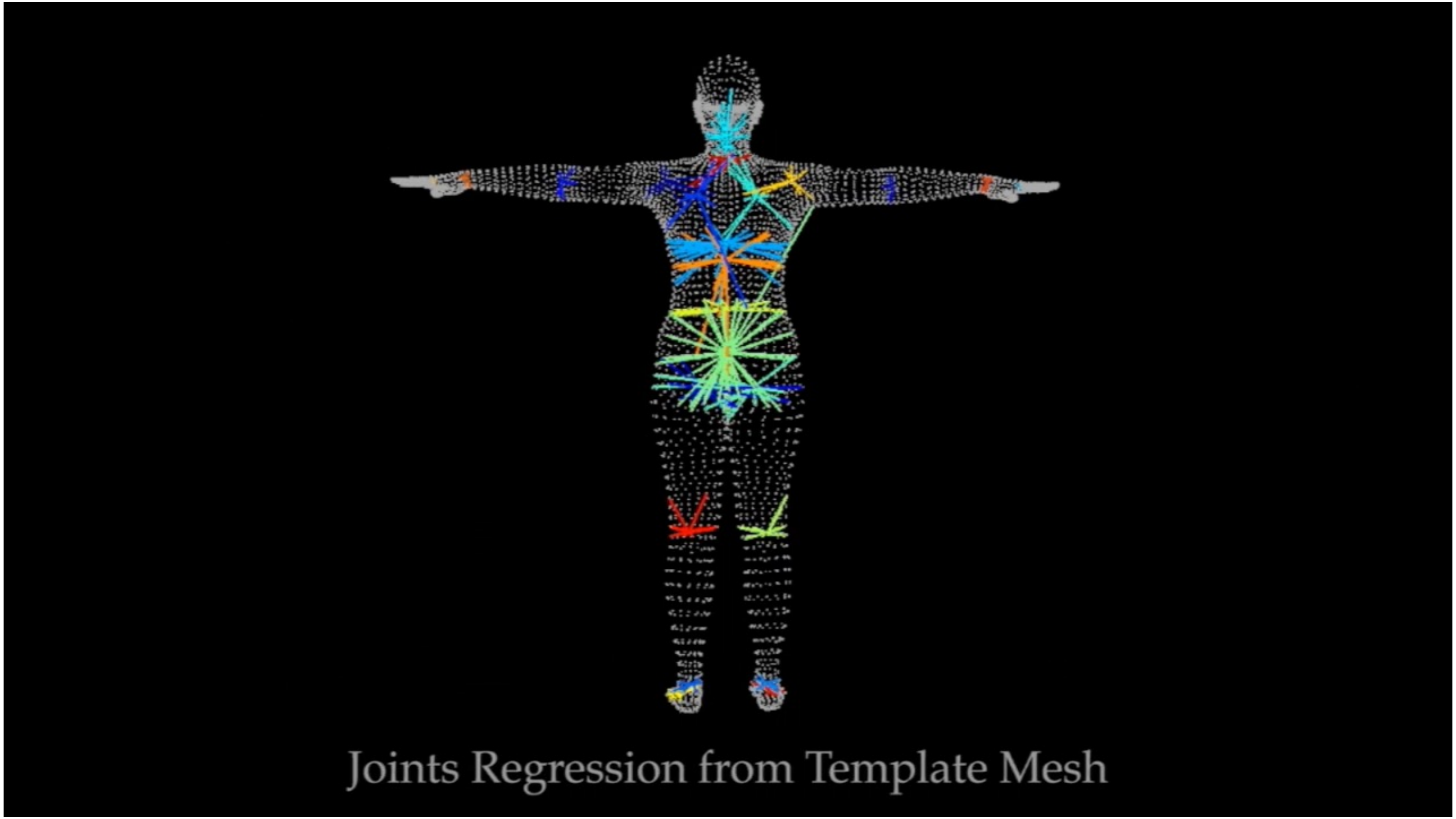
- How to get the joints $\mathbf{J}$ for a new shape? What is the simplest way?

- Joints are considered linear in rest vertices (much like in Allen et al. ’06)

$$\mathbf{J} = J(\mathbf{T}; \mathcal{J}) = \mathcal{J}\mathbf{T}$$

Joint regressor matrix

Joint Location Estimation



Adding a shape space

Problem: want a shape space with different identities

$$W(T(\vec{\theta}), J(\mathbf{T}), \mathcal{W}, \vec{\theta}) \mapsto \text{vertices}$$

$$T(\vec{\theta}) = \mathbf{T} + B_P(\vec{\theta})$$

Pose contribution

$$\left\{ B_P(\vec{\theta}) = \sum_{i}^{|f(\vec{\theta}|} f_i(\vec{\theta})\mathbf{P}_i \right.$$

Adding a shape space

Solution: add blend shapes linear with $\vec{\beta}$

$$W(T(\vec{\theta}, \vec{\beta}), J(\vec{\beta}), \mathcal{W}, \vec{\theta}) \mapsto \text{vertices}$$

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

Pose contribution $\left\{ B_P(\vec{\theta}) = \sum_i^{|f(\vec{\theta}|} f_i(\vec{\theta})\mathbf{P}_i \right.$

Shape contribution $\left\{ B_S(\beta) = \sum_j^{|\beta|} \beta_j S_j \right.$

SMPL

Additive Model

$$\bar{\mathbf{t}}'_i = \underbrace{\sum_{k=1}^{K} w_{k,i} G'_k(\vec{\theta}, J(\vec{\beta}))}_{\text{Joint locations}} (\underbrace{\bar{\mathbf{t}}_i}_{\text{Vertices}} + \overbrace{\mathbf{b}_{S,i}(\vec{\beta})}^{\text{Shape bs}} + \overbrace{\mathbf{b}_{P,i}(\vec{\theta})}^{\text{Pose bs}})$$

Parameterized Skinning

Standard skinning $W(\mathbf{T}, \mathbf{J}, \mathcal{W}, \vec{\theta}) \mapsto \text{vertices}$

SMPL model

$$M(\vec{\theta}, \vec{\beta}) = W(\mathbf{T}_F(\vec{\beta}, \theta), \mathbf{J}(\vec{\beta}), \mathcal{W}, \vec{\theta}) \mapsto \text{vertices}$$

SMPL is skinning parameterized by pose $\vec{\theta}$ and shape $\vec{\beta}$

SMPL

$$M(\underbrace{\overbrace{\vec{\theta}}^{\text{pose}}, \overbrace{\vec{\beta}}^{\text{shape}};}_{\text{Input}} \underbrace{\mathbf{T}, \mathcal{S}, \mathcal{P}, \mathcal{W}, \mathcal{J}}_{\text{Model parameters to be learned from data}})$$

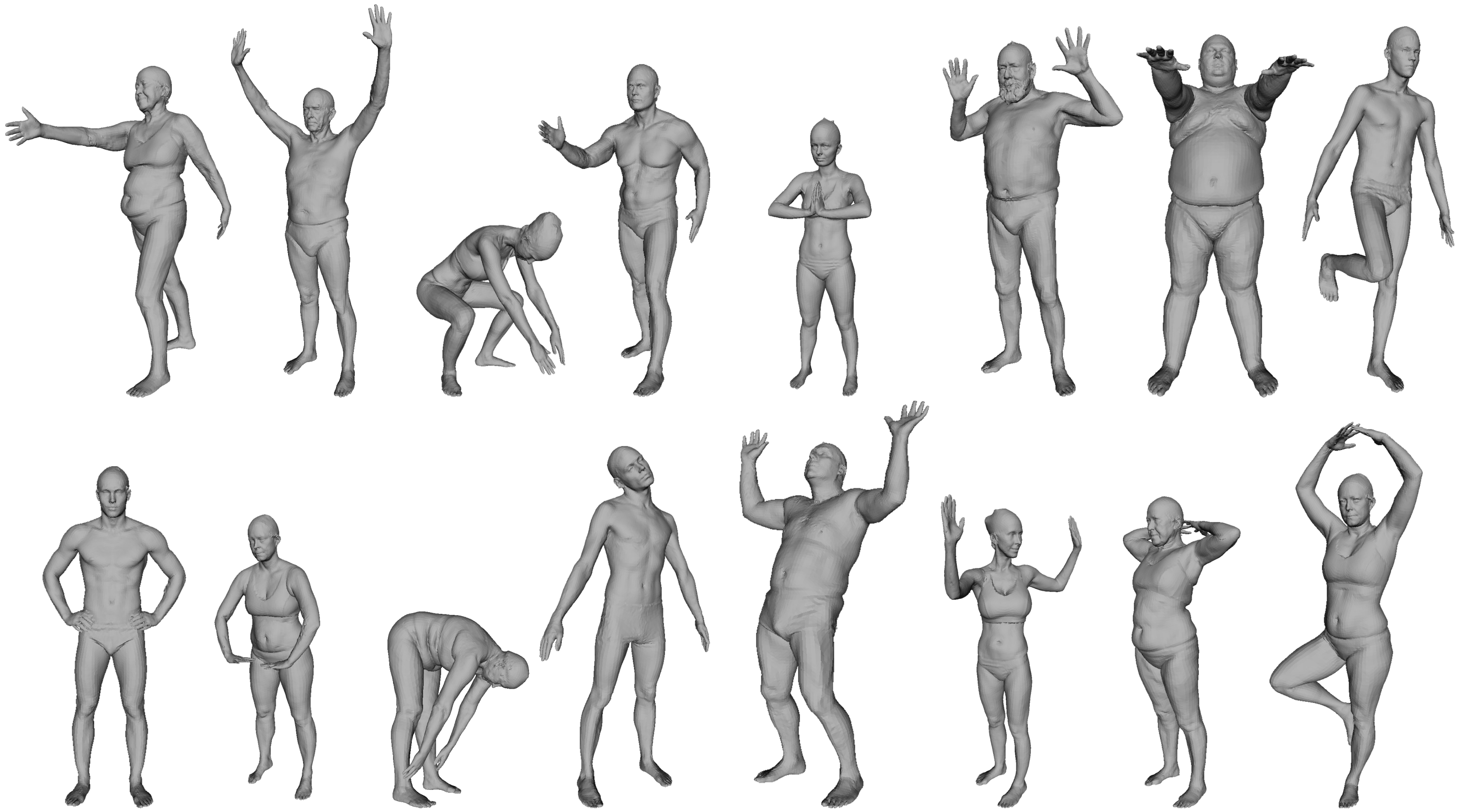
$$\left.\begin{array}{ll} \mathbf{T} & \text{Template (average shape)} \\ \mathcal{S} & \text{Shape blend shape matrix} \\ \mathcal{P} & \text{Pose blend shape matrix} \\ \mathcal{W} & \text{Blendweights matrix} \\ \mathcal{J} & \text{Joint regressor matrix} \end{array}\right\} \mathbf{w}$$

DATA

Model Training

Multipose database: 20 males, 24 females

1800 registrations

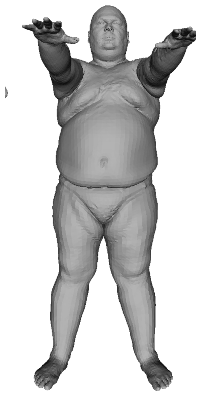


Model Training

Multishape database: PCA on ~2000 single-pose registrations per gender

Model Training

$$\mathbf{w} = \arg\min_{\mathbf{w}} \sum_{j} \| M(\vec{\theta}, \vec{\beta}; \mathbf{w}) -$$



$$\|^2$$

Training

$$\arg \underbrace{\min_{\mathbf{T},\mathcal{S},\mathcal{P},\mathcal{W},\mathcal{J}}} \sum_j \min_{\vec{\theta}_j,\vec{\beta}_j} \|M(\vec{\theta}_j, \vec{\beta}_j; \mathbf{T}, \mathcal{S}, \mathcal{P}, \mathcal{W}, \mathcal{J}) - \mathbf{V}_j\|^2$$

Model

Registrations

Ideally one wants to find the model parameters that minimize a single objective measuring the distance between **model** and **registrations**

Gradient based optimization!

Training Details

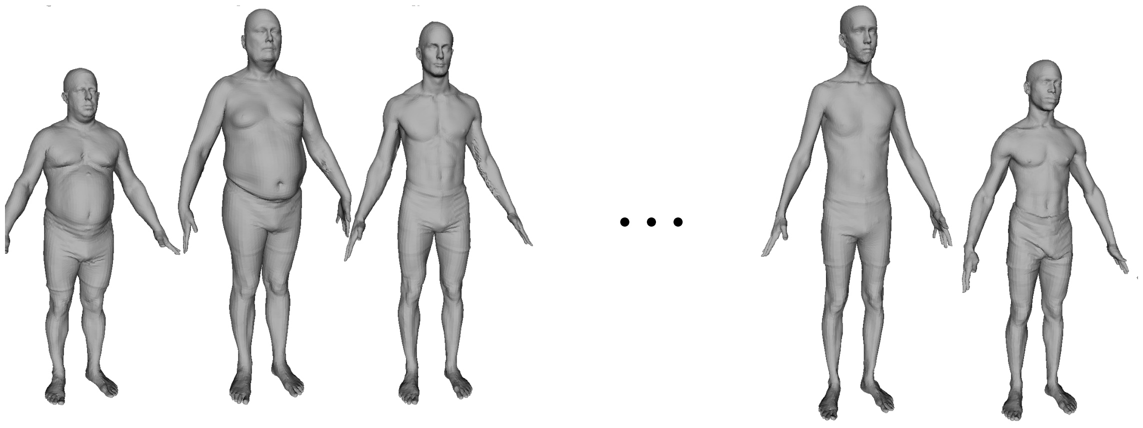
- $\mathcal{P}, \mathcal{W}, \mathcal{J}$ are trained from our **multipose** dataset
- $\mathcal{P}$ regularized towards zero (ridge regression)
- $\mathcal{W}$ regularized towards initialization
- $\mathcal{J}$ regularized towards predicting part boundary centers and is forced to be sparse
- $\mathbf{T}, \mathcal{S}$ are trained from our **multishape** dataset

Number of Parameters Learned

For a model with 6890 vertices:

- $\mathcal{P}$ 9x23x6890 = 4,278,690
- $\mathcal{W}$ 4x3x6890 = 82,680
- $\mathcal{J}$ 3x6890x23x3 = 1,426,230
- $\mathbf{T}, \mathcal{S}$ 3x6890 + 3x6890x10blendshapes = 227,370

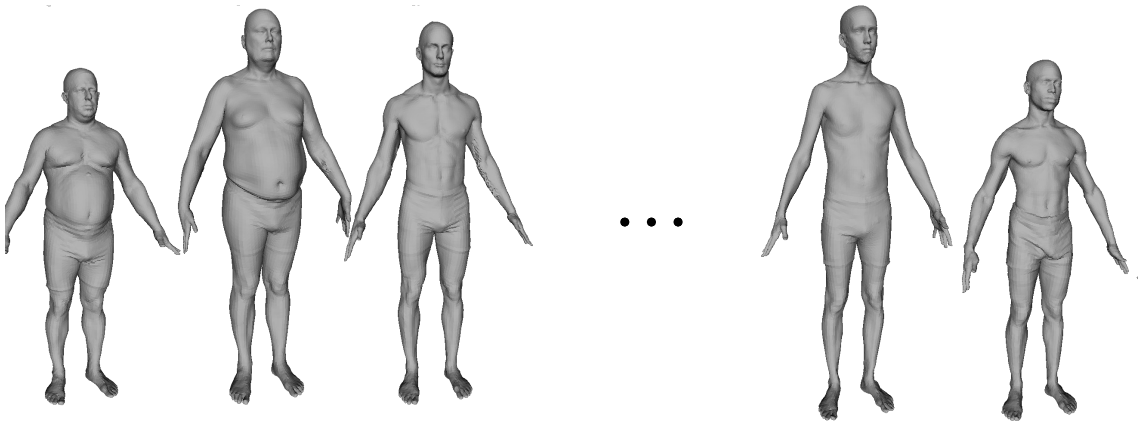
$\boxed{\text{A total of 6.014.970 parameters are learned}}$



$$\begin{bmatrix}\mathbf{V}_1 & \mathbf{V}_2 & \dots & \mathbf{V}_{N_{\mathrm{subj}}}\end{bmatrix} = \mathbf{T} + \begin{bmatrix}\mathbf{S}_1 & \mathbf{S}_2 & \dots & \mathbf{S}_{N_{\mathrm{subj}}}\end{bmatrix}\mathcal{B}$$

Average of shapes

Shape blend shapes are the first eigenvectors



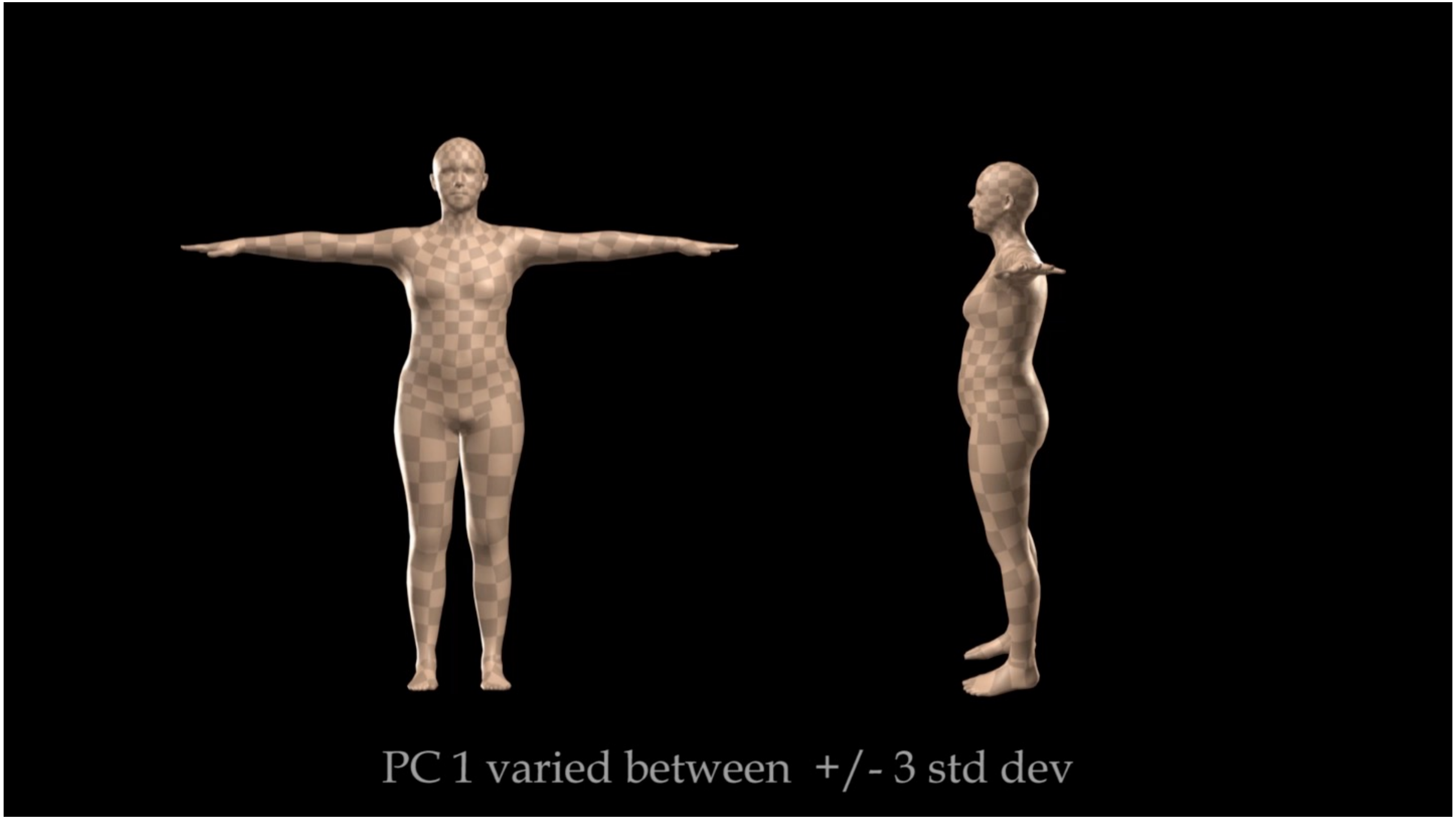
$$\begin{bmatrix} \mathbf{V}_1 & \mathbf{V}_2 & \dots & \mathbf{V}_{N_{\text{subj}}} \end{bmatrix} \approx \mathbf{T} + \mathcal{S}\mathcal{B}$$

Average of shapes

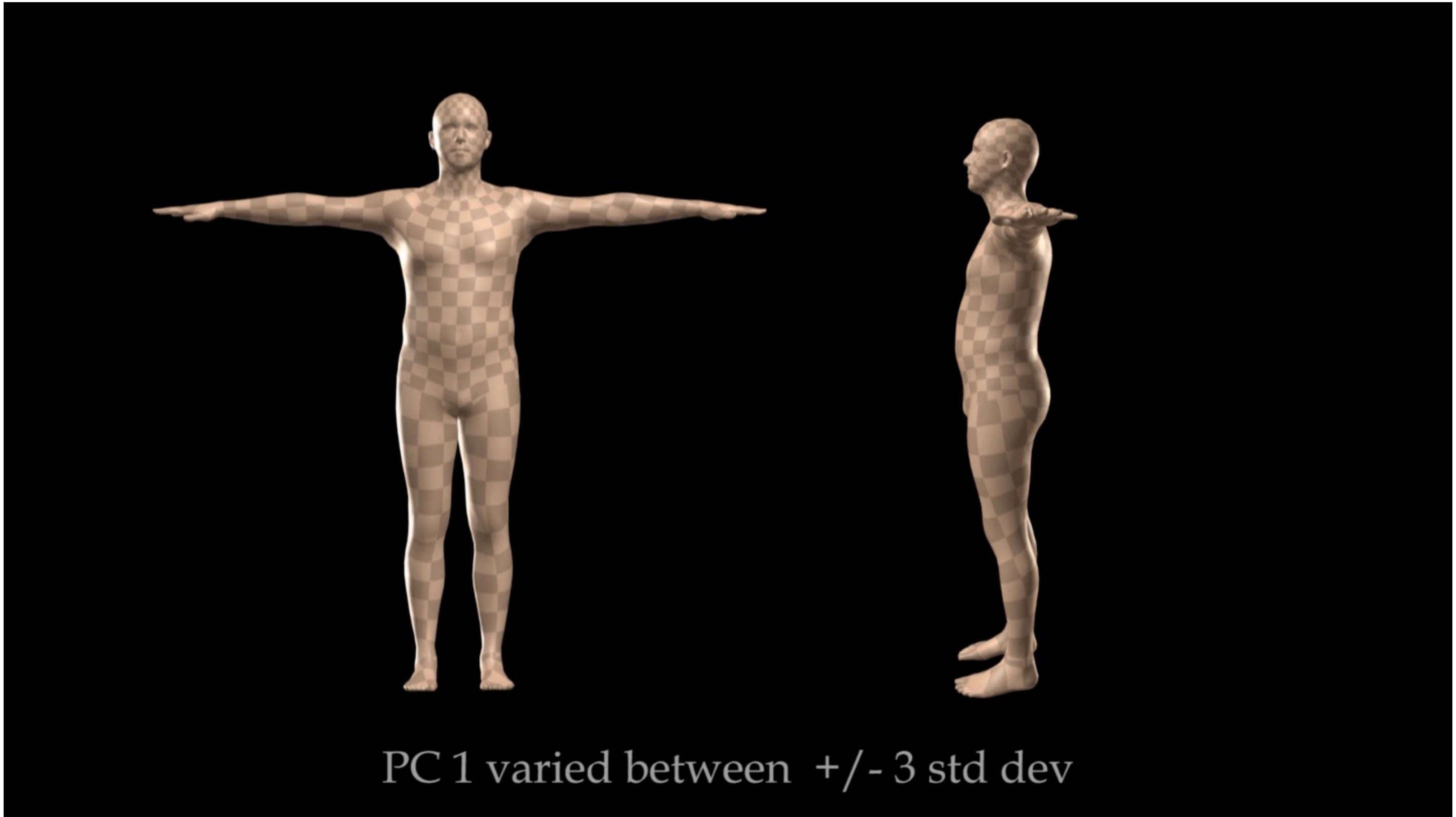
Shape blend shapes matrix

Before doing PCA all shapes have to be in the same pose
(pose needs to be optimized)

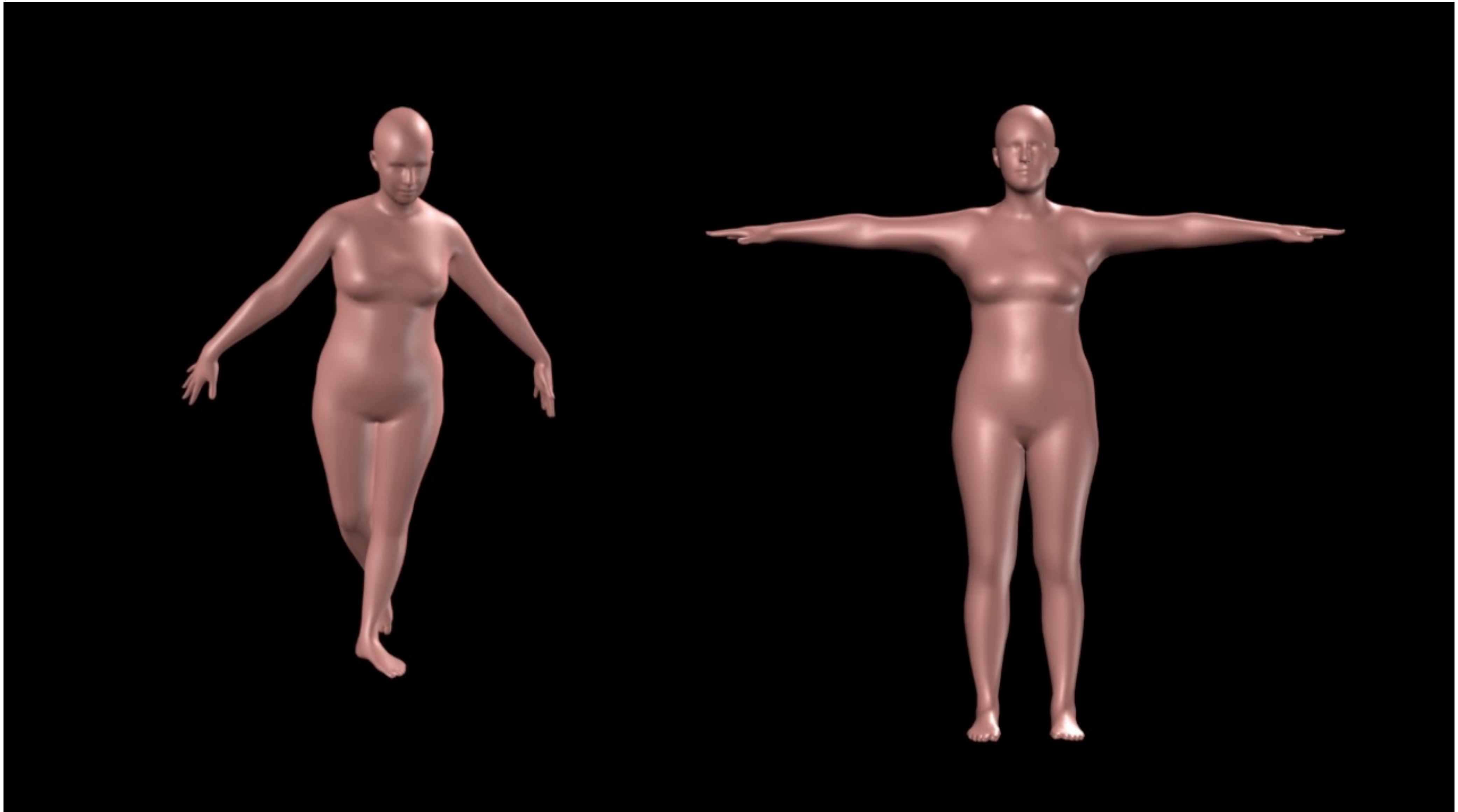
Shape Blend Shapes- Female



Shape Blend Shapes- Male



Pose Blendshapes



Two deformation models

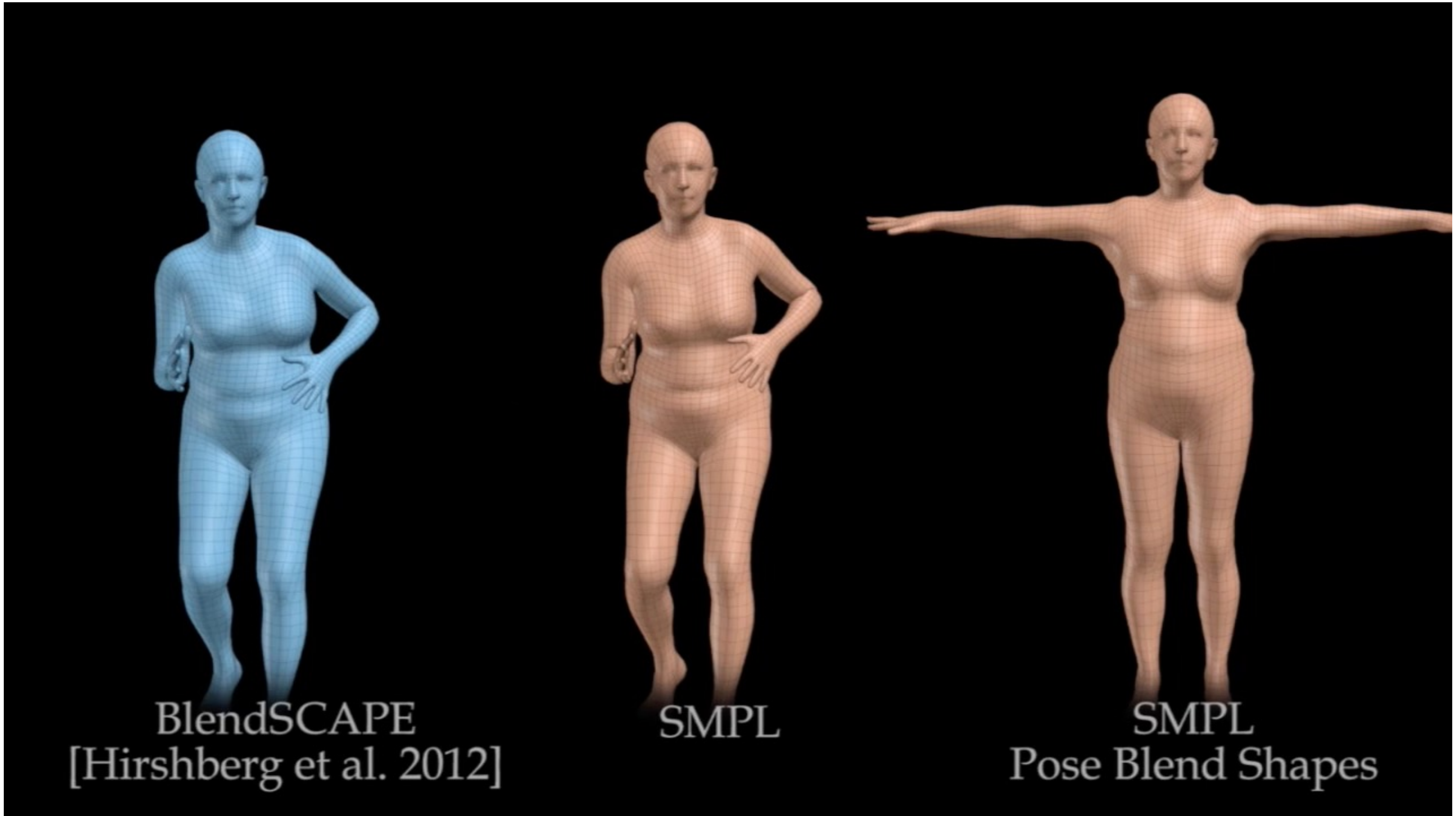
Local triangle deformations

- 3x3 transformations
- Applied to two edges per triangle
- No explicit center of rotation
- -> SCAPE

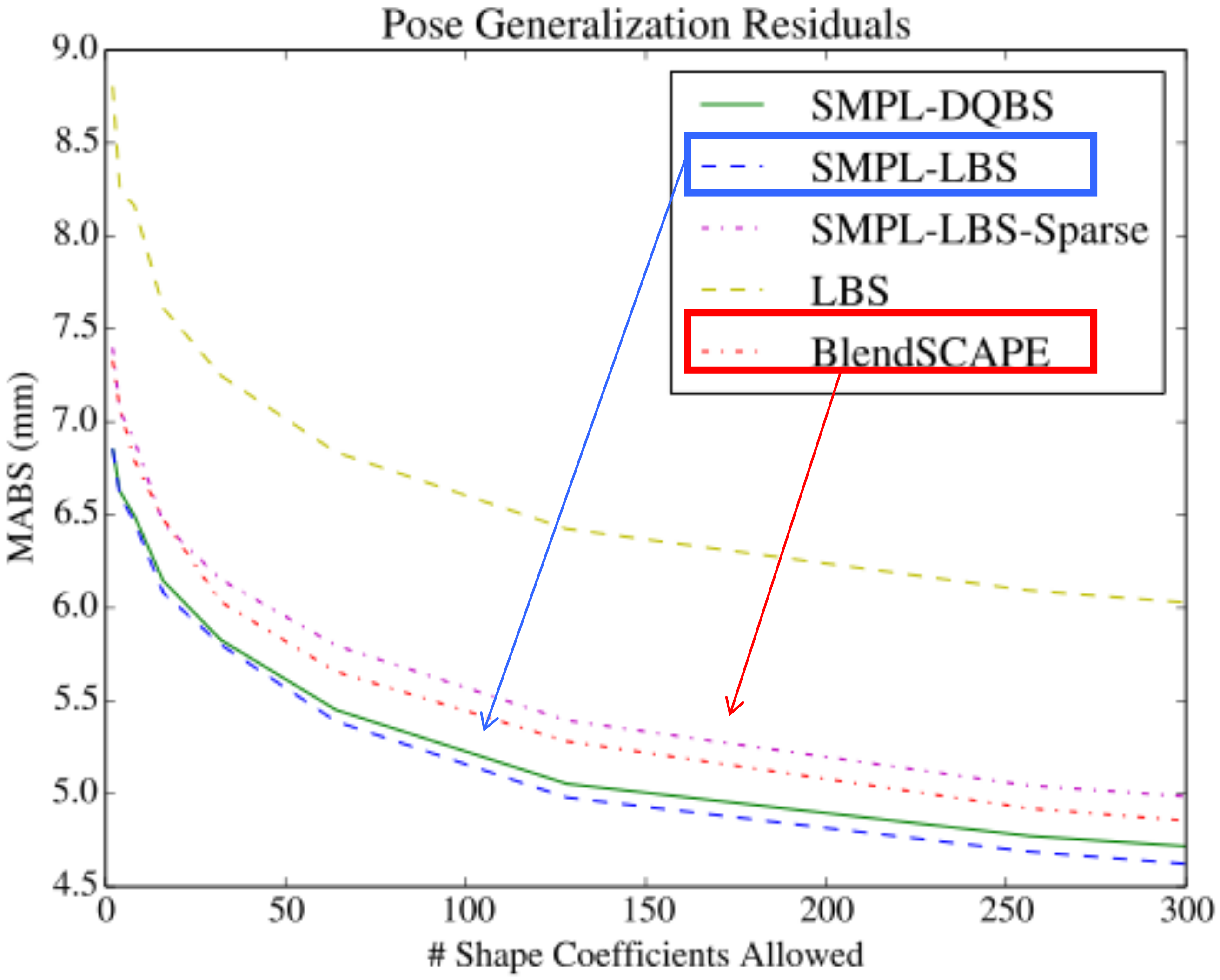
Global vertex deformations

- 3D displacements plus rigid body motion
- Applied to vertices
- Explicit center of rotation
- -> SMPL

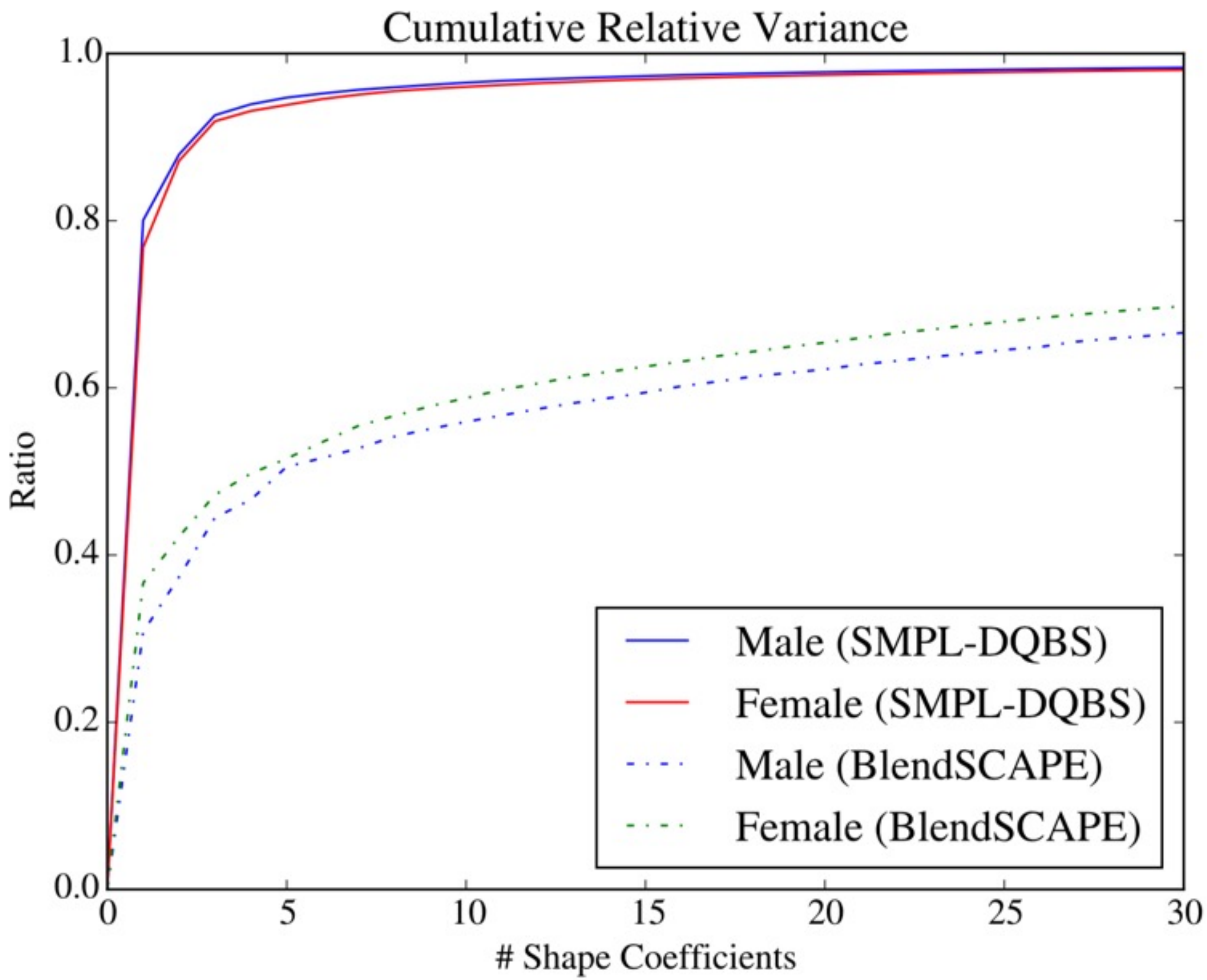
Comparison with BlendSCAPE



Comparison with BlendSCAPE



Comparison with BlendSCAPE

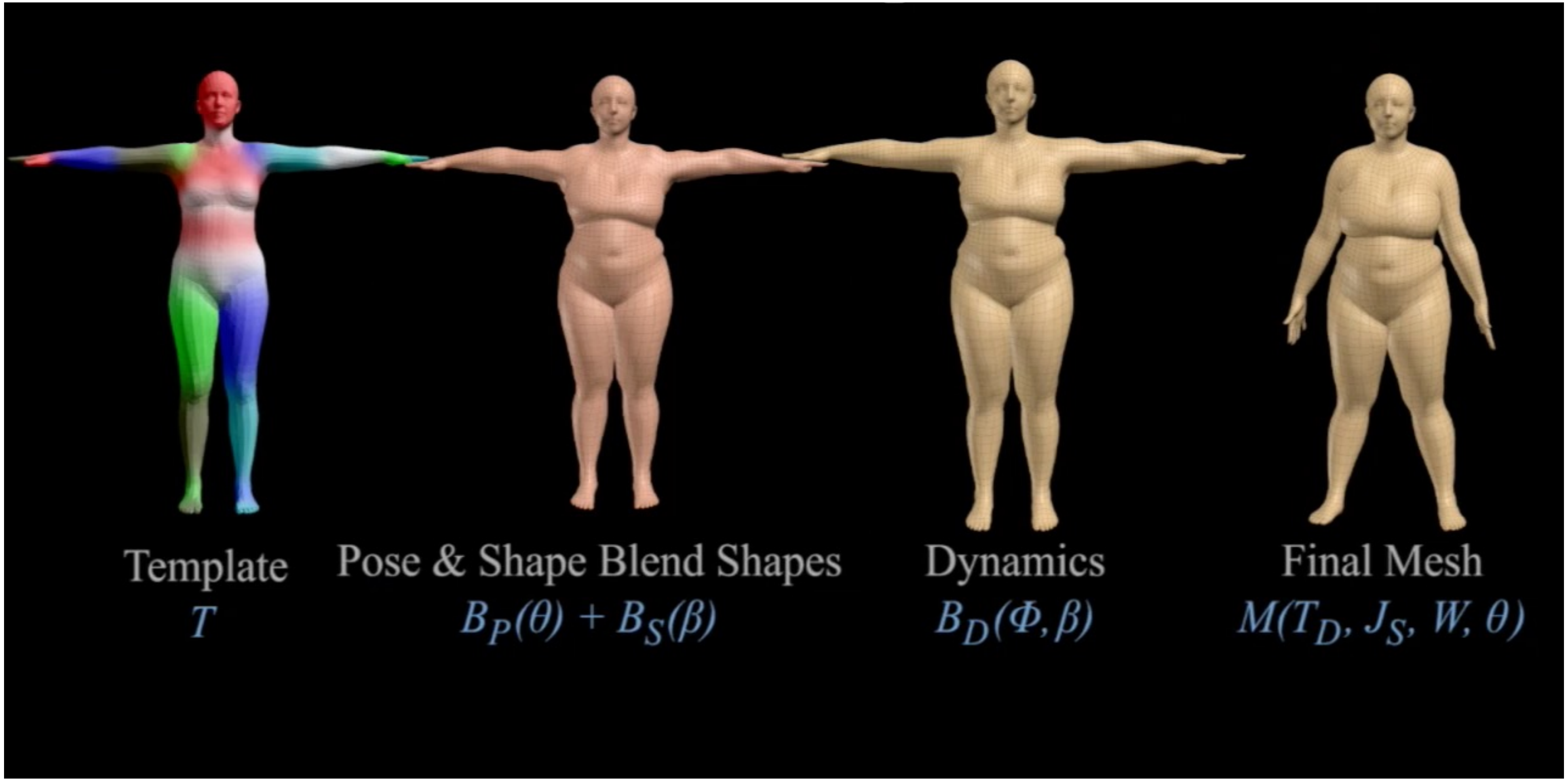


Conclusion

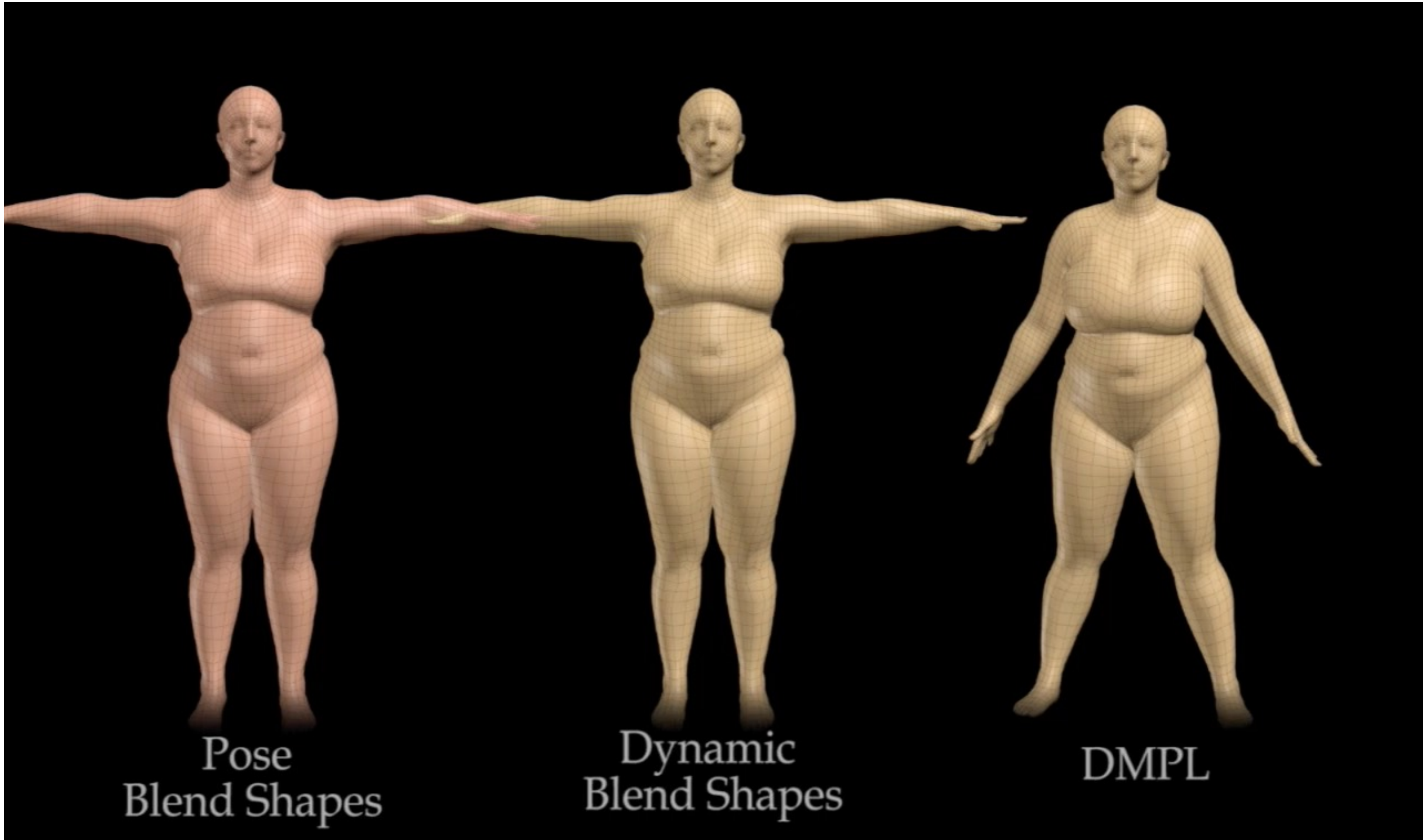
- **Speed**: fast run-time
- **Fidelity**: superior accuracy to Blend-SCAPE, trained on the same data
- **Compatibility**: works in Maya and Blender
- Is publicly available for research purposes

Download: http://smpl.is.tue.mpg.de

Model Decomposition

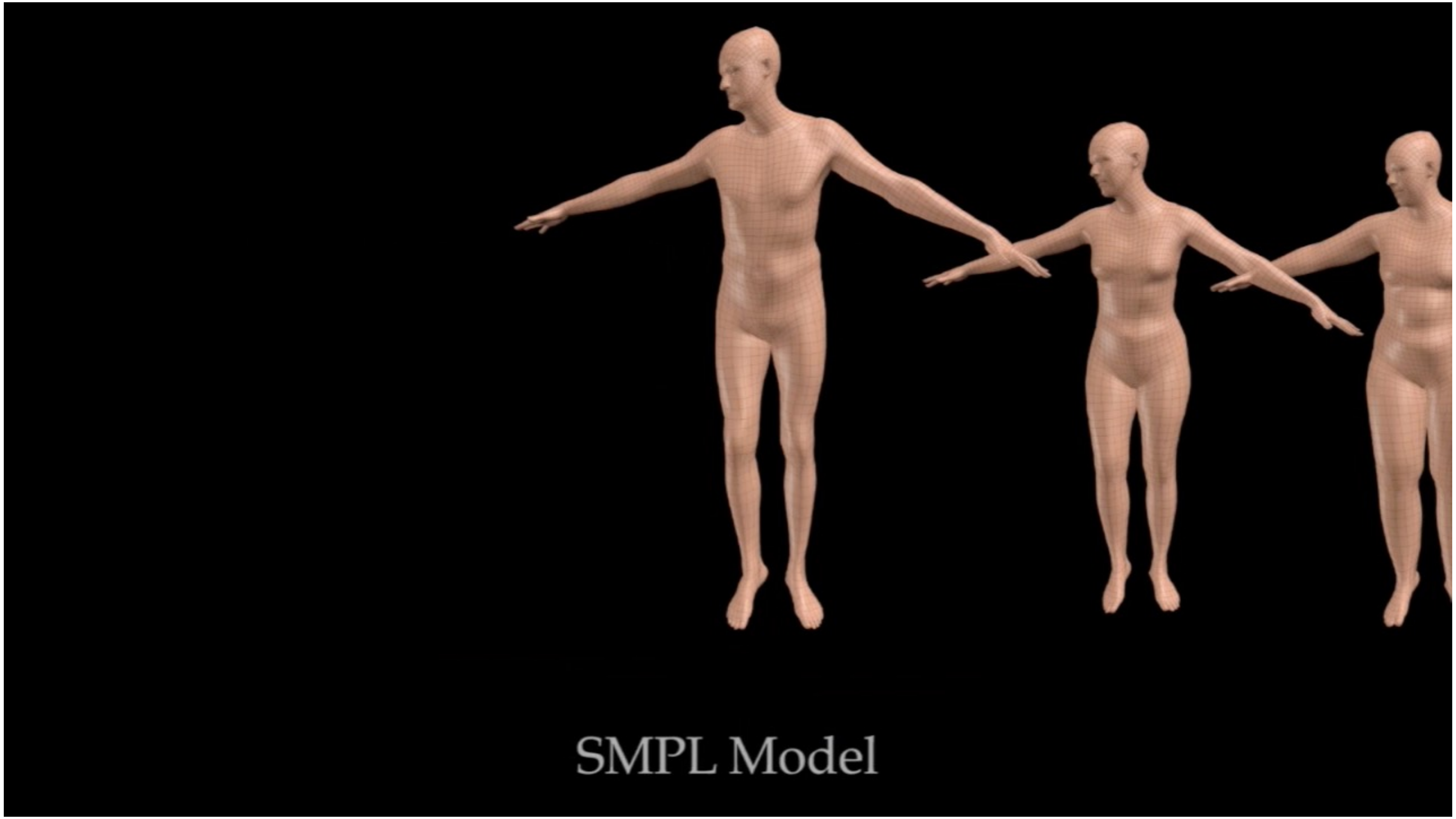


Dynamics of Soft Tissue



DMPL exaggeration

SMPL results



Slide credits and further reading

- Slides based on the Siggraph‘16 tutorial. Learning Human Bodies in Motion. (*Vertex Based Models)*

- Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, Michael J. BlackSMPL: A Skinned Multi-Person Linear Modelin ACM Trans. Graphics (Proc. SIGGRAPH Asia), vol. 34, no. 6, ACM, 248:1-248:16 2015.

- Javier Romero*, Dimitrios Tzionas* and Michael J Black. Embodied Hands: Modeling and Capturing Hands and Bodies Together *(Vertex Based for hands)*

- Gerard Pons-Moll, Javier Romero, Naureen Mahmood, Michael J. Black. Dyna: A Model of Dynamic Human Shape in Motion in ACM Transactions on Graphics, (Proc. SIGGRAPH), vol. 34, no. 4, ACM, 120:1-120:14 2015 *(Deformation gradients for soft-tissue)*

- Anguelov et al. SCAPE: Shape Completion and Animation of People. *(Deformation gradients)*