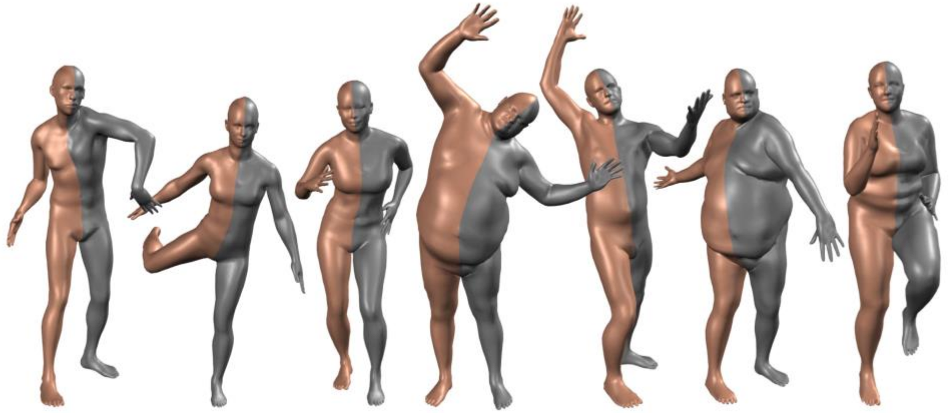
SMPL: A Skinned Multi-Person Linear Model

Thai Thanh Tuan, YoungSik Yun

Create: 6th July 2021



Introduction

- Paper: https://files.is.tue.mpg.de/black/papers/SMPL2015.pdf
- Project: https://smpl.is.tue.mpg.de/
- Github: SMPL-X https://github.com/vchoutas/smplx/tree/master/transfer_model

Introduction

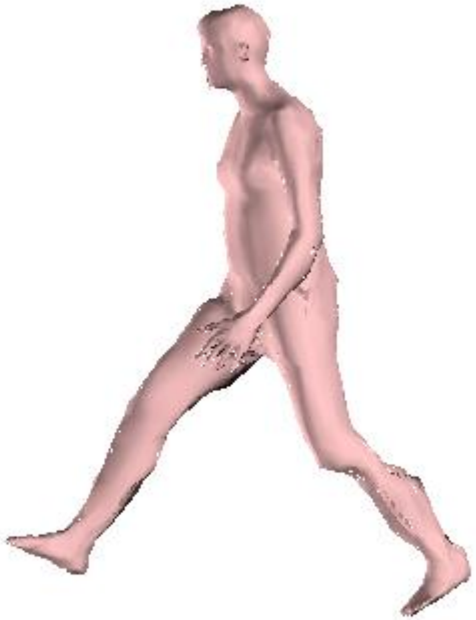
3D body
Dataset

+How to store and visualize?
→ **SMPL model**

Optimization based:
+SMPlify
+SMPLifyX

Deep learning:
+HMR
+SPIN
+Graph CNN based

Body and
shape
parameters



Store 3D data

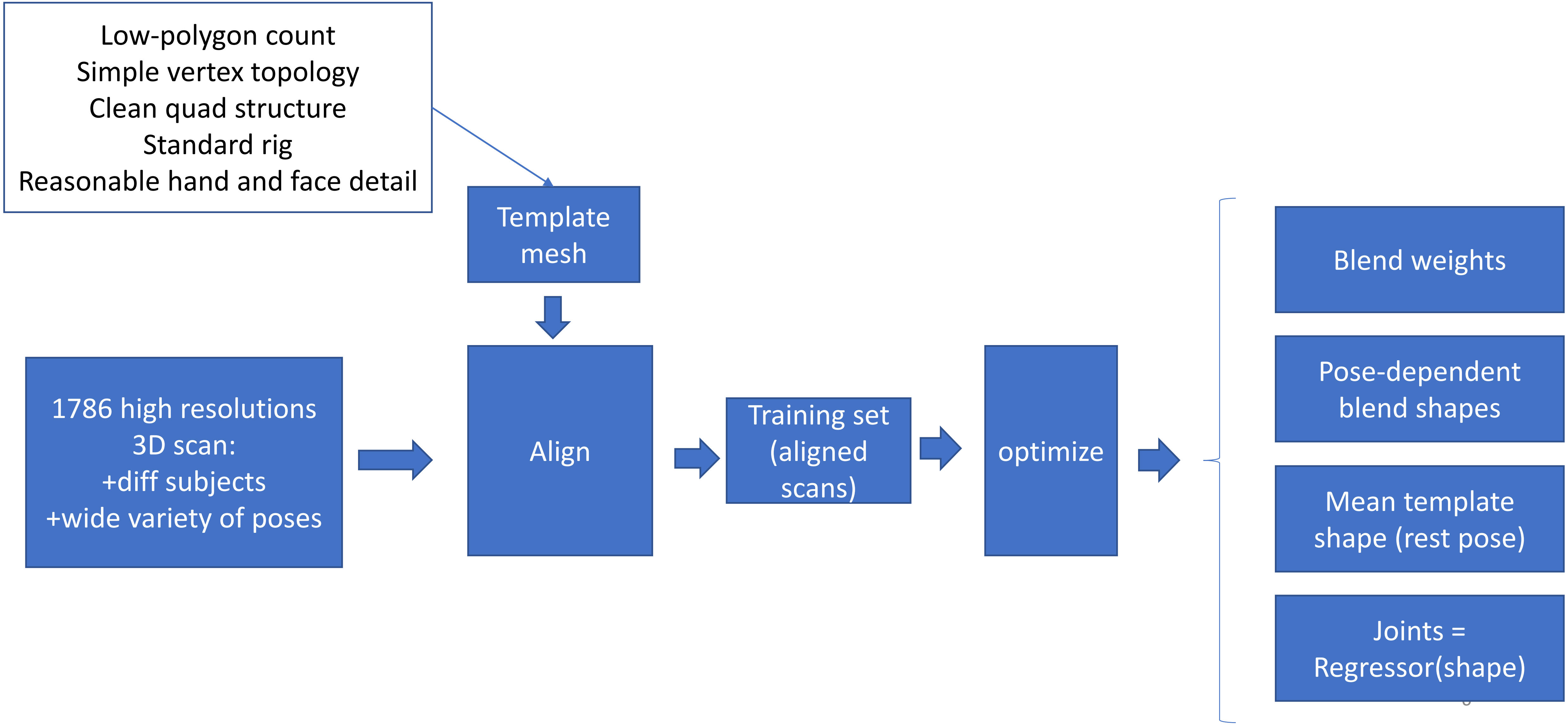
3D vertices coordinate:
+large memory
+not easy to change shape and pose

SMPL model
+Less memory
+Easy to change shape and pose
+Naturally/Realistic deform

Introduction

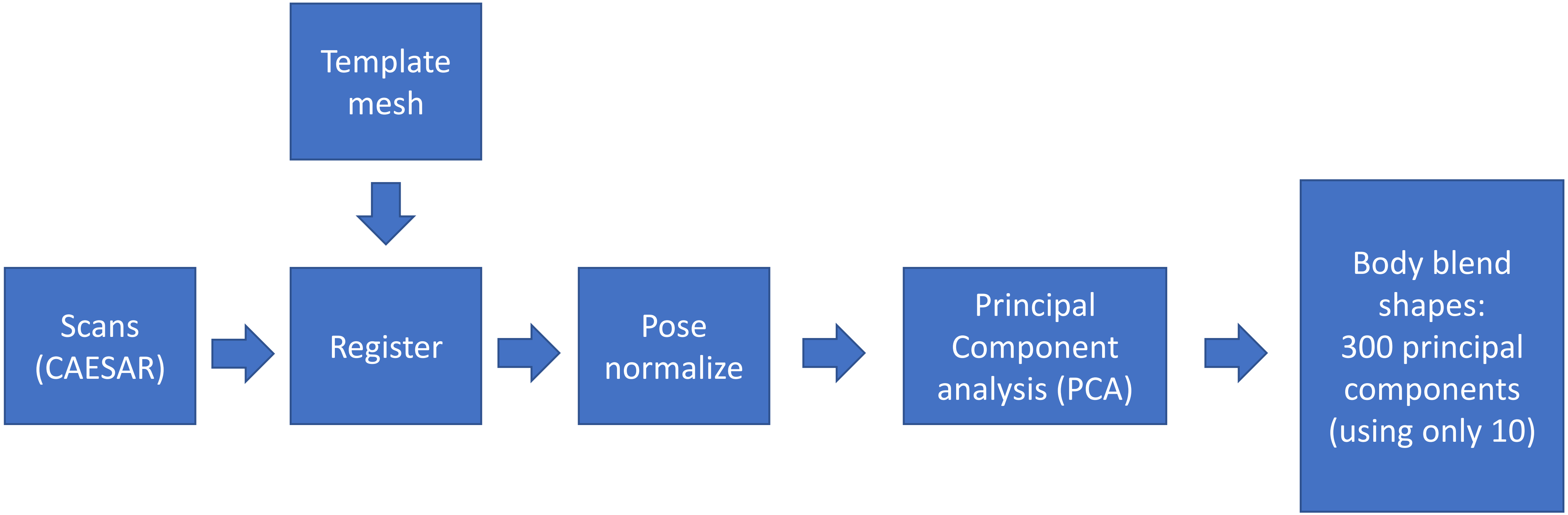
- Aim: Present a learned model of human body shape and pose dependent shape variation
  - More accurate
  - Comparable with existing graphics pipeline
- Parameters of the model(학습 데이터로부터 얻어옴)
  - rest pose template(zero pose:팔 벌린 상태에서 vertex의 위치 $\bar{T}$)
  - Blend weights($\mathcal{W}$ joint와 vertex간 연관관계의 척도)
  - Pose blendshape($\mathcal{P}$ template)
  - Shape(Identity blend shape($\mathcal{S}$ template)
  - Joint regressor matrix($\mathcal{J}$)

People deform with pose



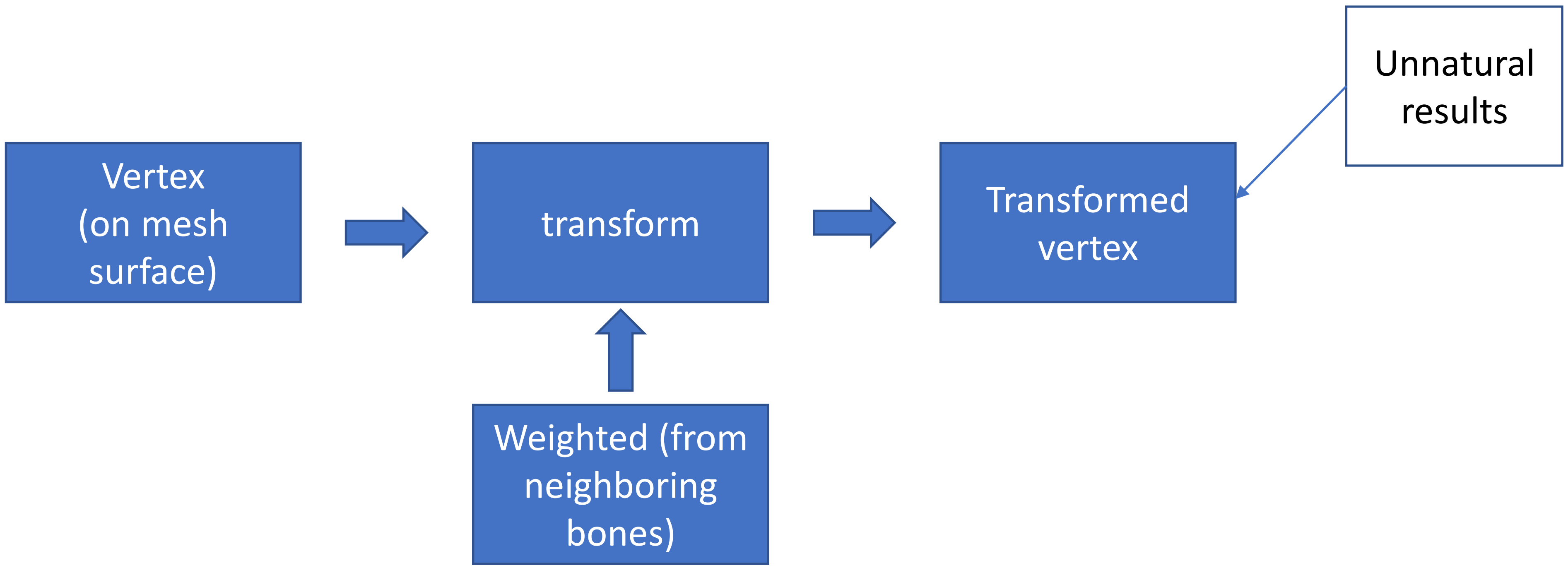
Linear models of male and female body shape

- CAESAR dataset [Robinette et al. 2002]
  - ~2000 scans / gender

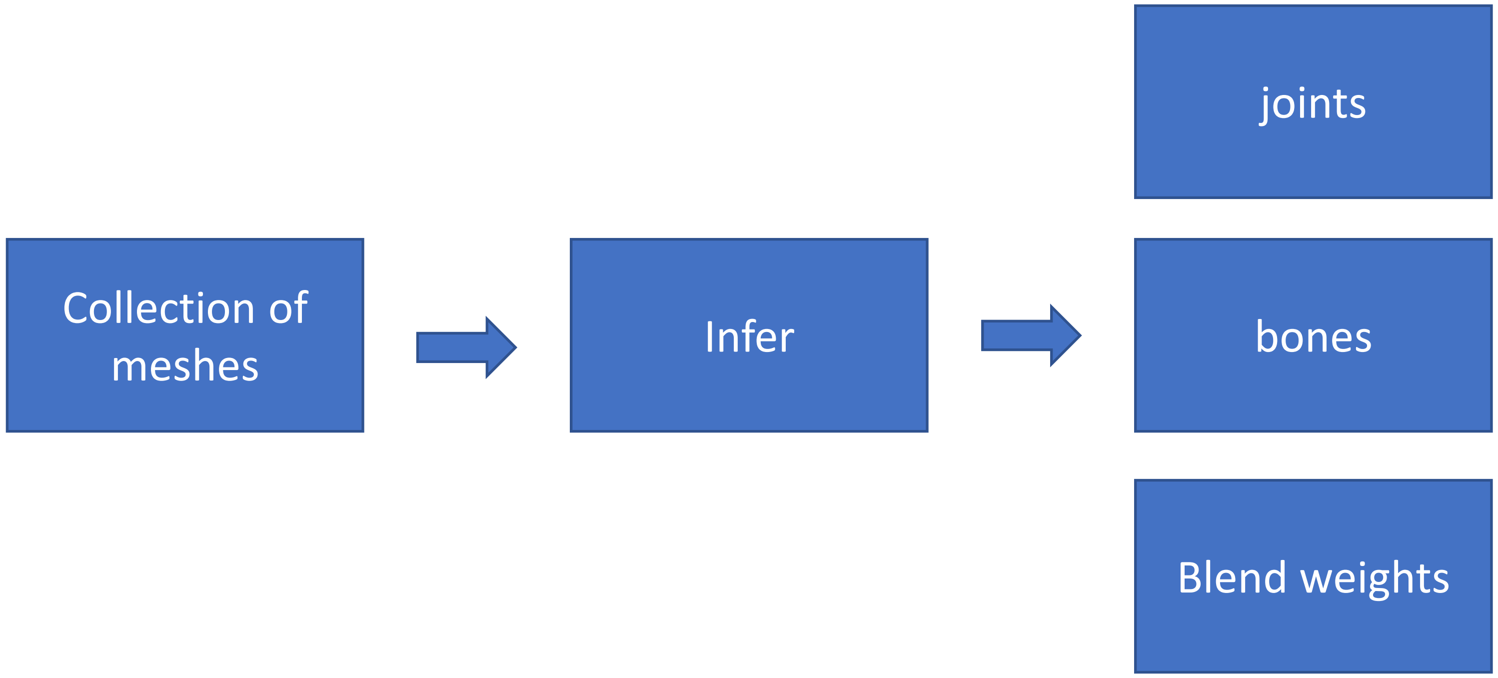


Related work: Blend skinning

- Skeleton subspace deformation methods
- Attach the surface of a mesh to an underlying skeletal structure



Related work: Auto rigging



Model formulation

- Mesh N = 6890 Vertices
- K = 23 joints
- A segmentation into parts
- Initial blend weights
- Skeletal rig

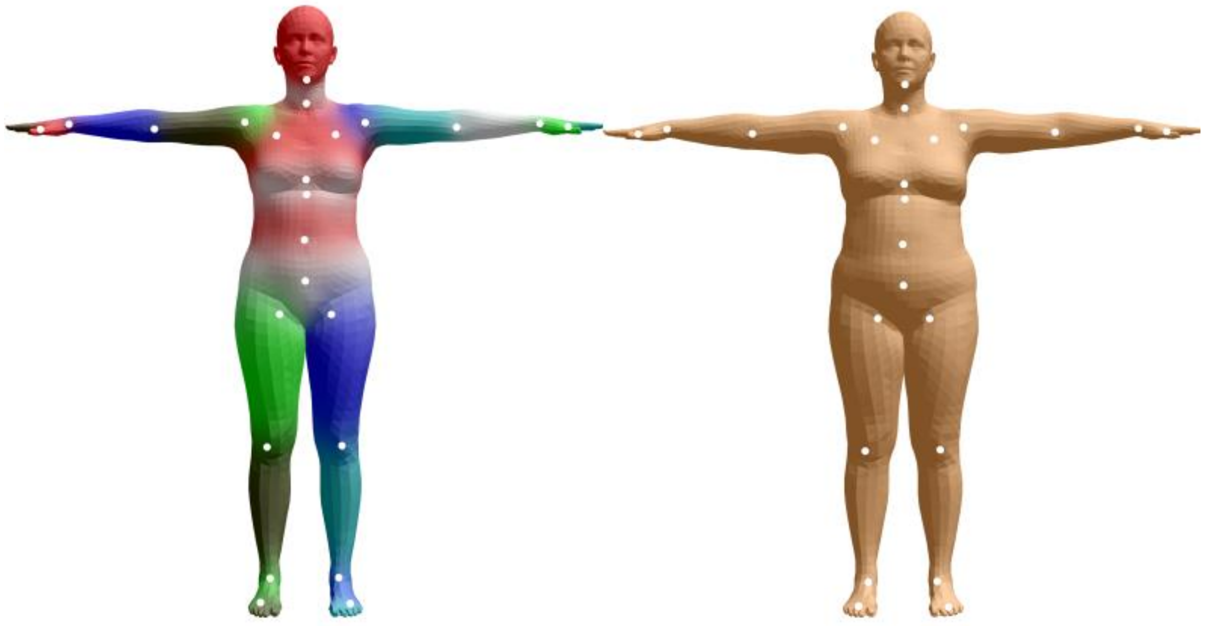
Mean template shape $\bar{\mathbf{T}} \in \mathbb{R}^{3N}$

- Zero pose: $\vec{\theta}^*$
- Blend weights: $\mathcal{W} \in \mathbb{R}^{N \times K}$

(a) $\bar{\mathbf{T}}$, $\mathcal{W}$

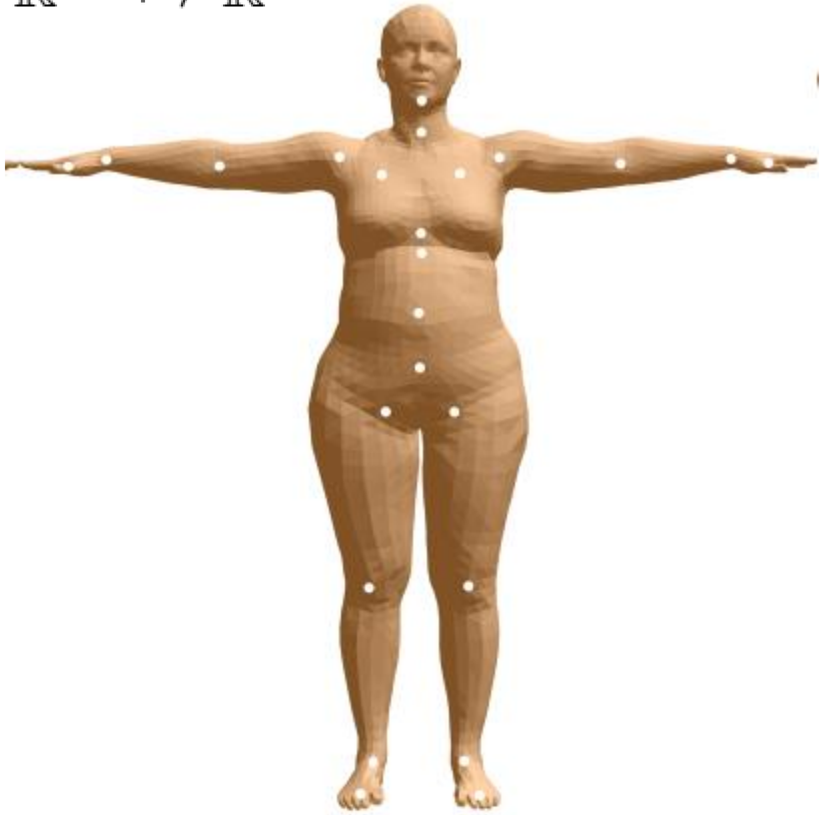
- A blend shape function: $B_S(\vec{\beta}) : \mathbb{R}^{|\vec{\beta}|} \mapsto \mathbb{R}^{3N}$
- Shape parameters: $\vec{\beta}$
- Output: blend shape sculpting the subject identity
- Function to predict K = 23 joint locations: white dots

$$J(\vec{\beta}) : \mathbb{R}^{|\vec{\beta}|} \mapsto \mathbb{R}^{3K}$$



(a) $\bar{\mathbf{T}}$, $\mathcal{W}$ **(b)** $\bar{\mathbf{T}} + B_S(\vec{\beta})$, $J(\vec{\beta})$

- Pose-dependent blend shape function: $B_P(\vec{\theta}) : \mathbb{R}^{|\vec{\theta}|} \mapsto \mathbb{R}^{3N}$
- Pose parameters: $\vec{\theta}$,



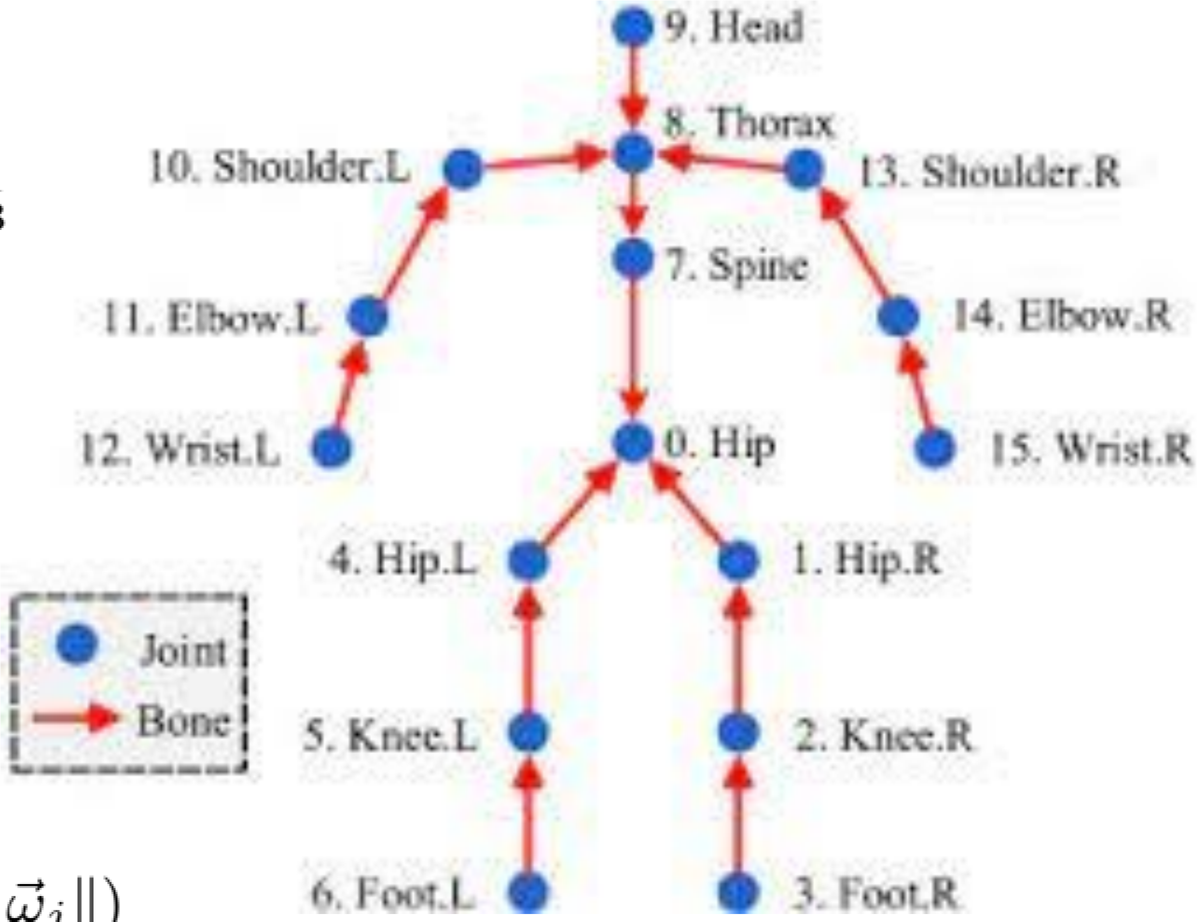
(c) $T_P(\vec{\beta}, \vec{\theta}) = \bar{\mathbf{T}} + B_S(\vec{\beta}) + B_P(\vec{\theta})$

- Standard blend skinning function $W(\cdot)$ (LBS or DQBS)
- Rotate the vertices around the estimated joint centers with smoothing
- Result: $M(\vec{\beta}, \vec{\theta}; \Phi) : \mathbb{R}^{|\vec{\theta}| \times |\vec{\beta}|} \mapsto \mathbb{R}^{3N}$
- Learned model parameters Φ

(d) $W(T_P(\vec{\beta}, \vec{\theta}), J(\vec{\beta}), \vec{\theta}, \mathcal{W})$

Blend skinning

- Particular vertex: $\mathbf{x}_i \in \mathbb{R}^3$
- Axis-angle between part k to its parents in the kinematic tree: $\vec{\omega}_k \in \mathbb{R}^3$
- K = 23 joints → $\vec{\theta} = [\vec{\omega}_0^T, \ldots, \vec{\omega}_K^T]^T$
- $|\vec{\theta}| = 3 \times 23 + 3 = 72$ parameters
- Unit norm axis of rotation: $\bar{\omega} = \frac{\vec{\omega}}{\|\vec{\omega}\|}$
- Each joint j → Rodrigues formula



$$\exp(\vec{\omega}_j) = \mathcal{I} + \widehat{\bar{\omega}}_j \sin(\|\vec{\omega}_j\|) + \widehat{\bar{\omega}}_j^2 \cos(\|\vec{\omega}_j\|)$$

$\widehat{\bar{\omega}}$ is the skew symmetric matrix of the 3-vector $\bar{\omega}$

Blend skinning

- Standard linear blend skinning function

$$W(\bar{\mathbf{T}}, \mathbf{J}, \vec{\theta}, \mathcal{W}) \,:\, \mathbb{R}^{3N \times 3K \times |\vec{\theta}| \times |\mathcal{W}|} \mapsto \mathbb{R}^{3N}$$

- Rest pose: $\bar{\mathbf{T}}$
- Joint locations: **J**
- A pose: $\vec{\theta} = [\vec{\omega}_0^T, \ldots, \vec{\omega}_K^T]^T$
- Blend weights: $\mathcal{W}$
- Return: The posed vertices
- Each vertex $\bar{\mathbf{t}}_i$ in $\bar{\mathbf{T}}$ →transformed $\bar{\mathbf{t}}'_i$

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

$$G'_k(\vec{\theta}, \mathbf{J}) = G_k(\vec{\theta}, \mathbf{J}) G_k(\vec{\theta}^*, \mathbf{J})^{-1}$$

$$G_k(\vec{\theta}, \mathbf{J}) = \prod_{j \in A(k)} \left[\begin{array}{c|c} \exp(\vec{\omega}_j) & \mathbf{j}_j \\ \hline \vec{0} & 1 \end{array} \right]$$

Blend skinning

- Element of blend weight matrix $w_{k,i}$
  - How much the rotation of part k effect vertex i
- 3x3 rotation matrix $\exp(\vec{\theta}_j)$ corresponding to join j.
- World transformation of joints k: $G_k(\vec{\theta}, \mathbf{J})$
- →remove transformation of rest pose $\vec{\theta}^*$:

$$G'_k(\vec{\theta}, \mathbf{J})$$

- Joints j location: $\mathbf{j}_j$
- Ordered set of joint ancestors of join k: $A(k)$

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

$$G'_k(\vec{\theta}, \mathbf{J}) = G_k(\vec{\theta}, \mathbf{J}) G_k(\vec{\theta}^*, \mathbf{J})^{-1}$$

$$G_k(\vec{\theta}, \mathbf{J}) = \prod_{j \in A(k)} \left[\begin{array}{c|c} \exp(\vec{\omega}_j) & \mathbf{j}_j \\ \hline \vec{0} & 1 \end{array} \right]$$

Blend skinning

- SMPL blend skinning model: $M(\vec{\beta}, \vec{\theta}; \Phi)$

$$\begin{aligned} M(\vec{\beta}, \vec{\theta}) &= W(T_P(\vec{\beta}, \vec{\theta}), J(\vec{\beta}), \vec{\theta}, \mathcal{W}) \\ T_P(\vec{\beta}, \vec{\theta}) &= \bar{\mathbf{T}} + B_S(\vec{\beta}) + B_P(\vec{\theta}) \end{aligned}$$

- Vectors of vertices representing offsets from the template.
  - Shape blend shape $B_S(\vec{\beta})$
  - Pose blend shape: $B_P(\vec{\theta})$

Explain in next slides

- Then:

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

Shape blend shapes

- Body shapes of different people represented by a linear function

$$B_S(\vec{\beta}; \mathcal{S}) = \sum_{n=1}^{|\vec{\beta}|} \beta_n \mathbf{S}_n$$

$$\vec{\beta} = [\beta_1, . |\vec{\beta}| \, \beta_{|\vec{\beta}|}]^T$$

- Number of linear shape coefficients: $|\vec{\beta}|$
- Orthonormal principal components of shape displacements: $\mathbf{S}_n \in \mathbb{R}^{3N}$
- S: learned from registered training meshes, Sec. 4

Pose blend shapes

- Maps function: $R : \mathbb{R}^{|\vec{\theta}|} \mapsto \mathbb{R}^{9K}$
- Vector of concatenated part relative rotation matrices: $R(\vec{\theta})$: 23 x 9, non-linear with $\vec{\theta}$. (functions of sines and cosines)
- $R^*(\vec{\theta}) = (R(\vec{\theta}) - R(\vec{\theta}^*))$ where $\vec{\theta}^*$ denotes the rest pose.

$$B_P(\vec{\theta};\mathcal{P}) = \sum_{n=1}^{9K} (R_n(\vec{\theta}) - R_n(\vec{\theta}^*))\mathbf{P}_n$$

$\mathbf{P}_n \in \mathbb{R}^{3N}$,

learned from registered training meshes, Sec. 4

- Matrix of all 207 pose blend shapes: $\mathcal{P} = [\mathbf{P}_1, \ldots, \mathbf{P}_{9K}] \in \mathbb{R}^{3N \times 9K}$
- In rest pose, pose blend shapes is zero.

Joint locations

- Joints as a function of the body shape, $\vec{\beta},$

$$J(\vec{\beta}; \mathcal{J}, \bar{\mathbf{T}}, \mathcal{S}) = \mathcal{J}(\bar{\mathbf{T}} + B_S(\vec{\beta}; \mathcal{S}))$$

- Matrix that transforms rest vertices into rest joints: $\mathcal{J}$
  - Models which mesh vertices are important
  - How to combine them to estimate the joint locations.

SMPL model

- SMPL model:

$$\Phi = \{\bar{\mathbf{T}}, \mathcal{W}, \mathcal{S}, \mathcal{J}, \mathcal{P}\}$$

- How to lean these in Sec. 4
- After learned, SMPL model:

$$M(\vec{\beta}, \vec{\theta}; \Phi) = W\left(T_P(\vec{\beta}, \vec{\theta}; \bar{\mathbf{T}}, \mathcal{S}, \mathcal{P}), J(\vec{\beta}; \mathcal{J}, \bar{\mathbf{T}}, \mathcal{S}), \vec{\theta}, \mathcal{W}\right)$$

- Vertex transformed:

$$\mathbf{t}'_i = \sum_{k=1}^{K} w_{k,i} G'_k(\vec{\theta}, J(\vec{\beta}; \mathcal{J}, \bar{\mathbf{T}}, \mathcal{S}))\mathbf{t}_{P,i}(\vec{\beta}, \vec{\theta}; \bar{\mathbf{T}}, \mathcal{S}, \mathcal{P})$$

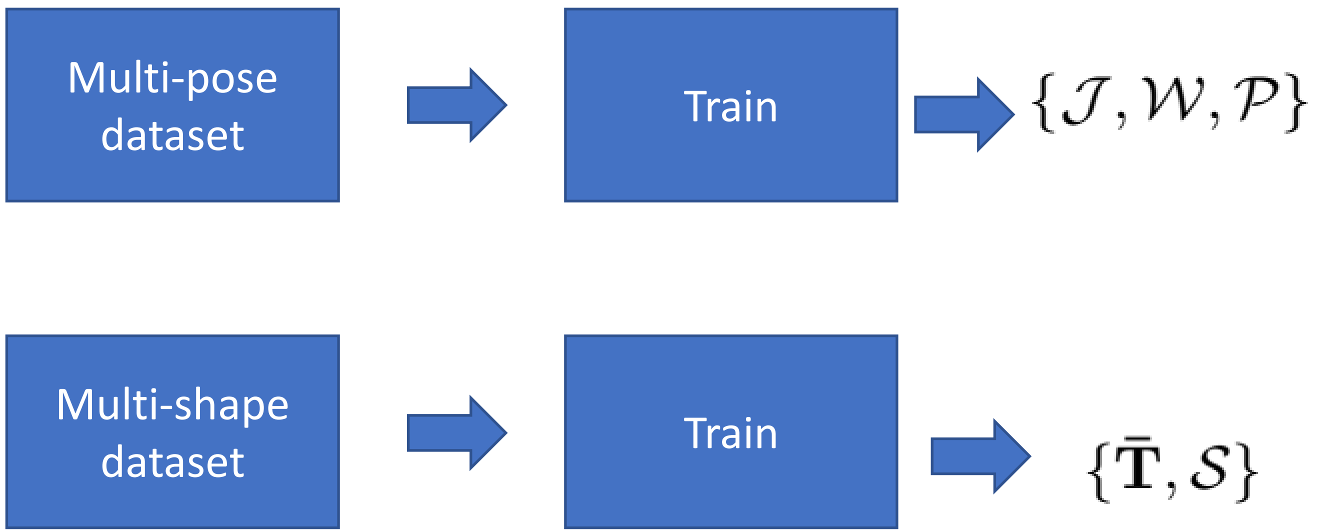
- Vertex i after applying the blend shapes

$$\mathbf{t}_{P,i}(\vec{\beta}, \vec{\theta}; \bar{\mathbf{T}}, \mathcal{S}, \mathcal{P}) = \bar{\mathbf{t}}_i + \sum_{m=1}^{|\vec{\beta}|} \beta_m \mathbf{s}_{m,i} + \sum_{n=1}^{9K} (R_n(\vec{\theta}) - R_n(\vec{\theta}^*))\mathbf{p}_{n,i}$$

4. Training

- Aligned meshes: “registrations“
- Multi-pose dataset:
  - 1786 registrations of 40 individuals
  - 891 of 20 females
  - 895 of 20 males
- Multi-shape dataset: registrations to the CAESAR dataset
  - 1700 registration for males
  - 2100 for females
- jth mesh in multi-pose dataset: $\mathbf{V}_j^P$; and in multi-shape dataset: $\mathbf{V}_j^S$
- Minimize vertex reconstruction error

Training


Multi-pose dataset
Train
$\{\mathcal{J}, \mathcal{W}, \mathcal{P}\}$
Multi-shape dataset
Train
$\{\bar{\mathbf{T}}, \mathcal{S}\}$

Pose parameter training

Compute rest template???

$\hat{\mathbf{T}}_i^P$

Multi-pose dataset

Compute joint location for each subject i

$\hat{\mathbf{J}}_i^P$

P_{reg}

Number of meshes

P_{subj}

Compute pose parameters for each registration j

$\vec{\theta}_j$

Number of subjects

$$E_D(\hat{\mathbf{T}}^P, \hat{\mathbf{J}}^P, \mathcal{W}, \mathcal{P}, \Theta) = \sum_{j=1}^{P_{\text{reg}}} ||\mathbf{V}_j^P - W(\hat{\mathbf{T}}_{s(j)}^P + B_P(\vec{\theta}_j; \mathcal{P}), \hat{\mathbf{J}}_{s(j)}^P, \vec{\theta}_j, \mathcal{W})||^2$$

$$\Theta = \left\{\vec{\theta}_1, \ldots, \vec{\theta}_{P_{\text{reg}}}\right\}$$

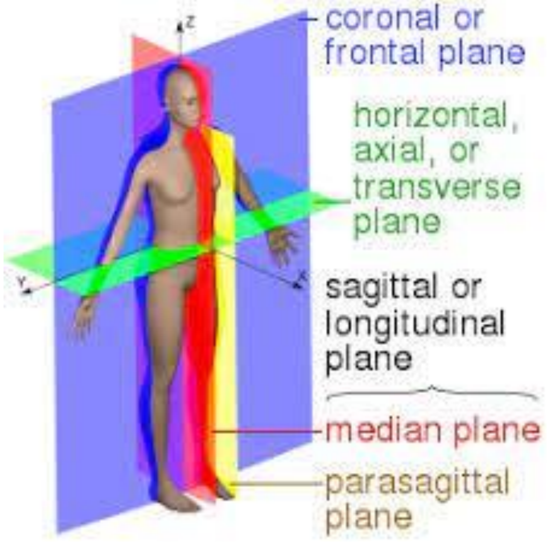
$$\hat{\mathbf{T}}^P = \{\hat{\mathbf{T}}_i^P\}_{i=1}^{P_{\text{subj}}}$$

All rest poses

$$\hat{\mathbf{J}}^P = \{\hat{\mathbf{J}}_i^P\}_{i=1}^{P_{\text{subj}}}$$

All rest joints

Pose parameter training



- Regularization term:
  - Penalizes left-right asymmetry $\hat{\mathbf{J}}^P$ and $\hat{\mathbf{T}}^P$

$$E_Y(\hat{\mathbf{J}}^P, \hat{\mathbf{T}}^P) = \sum_{i=1}^{P_{\text{subj}}} \lambda_U ||\hat{\mathbf{J}}_i^P - U(\hat{\mathbf{J}}_i^P)||^2 + ||\hat{\mathbf{T}}_i^P - U(\hat{\mathbf{T}}_i^P)||^2,$$

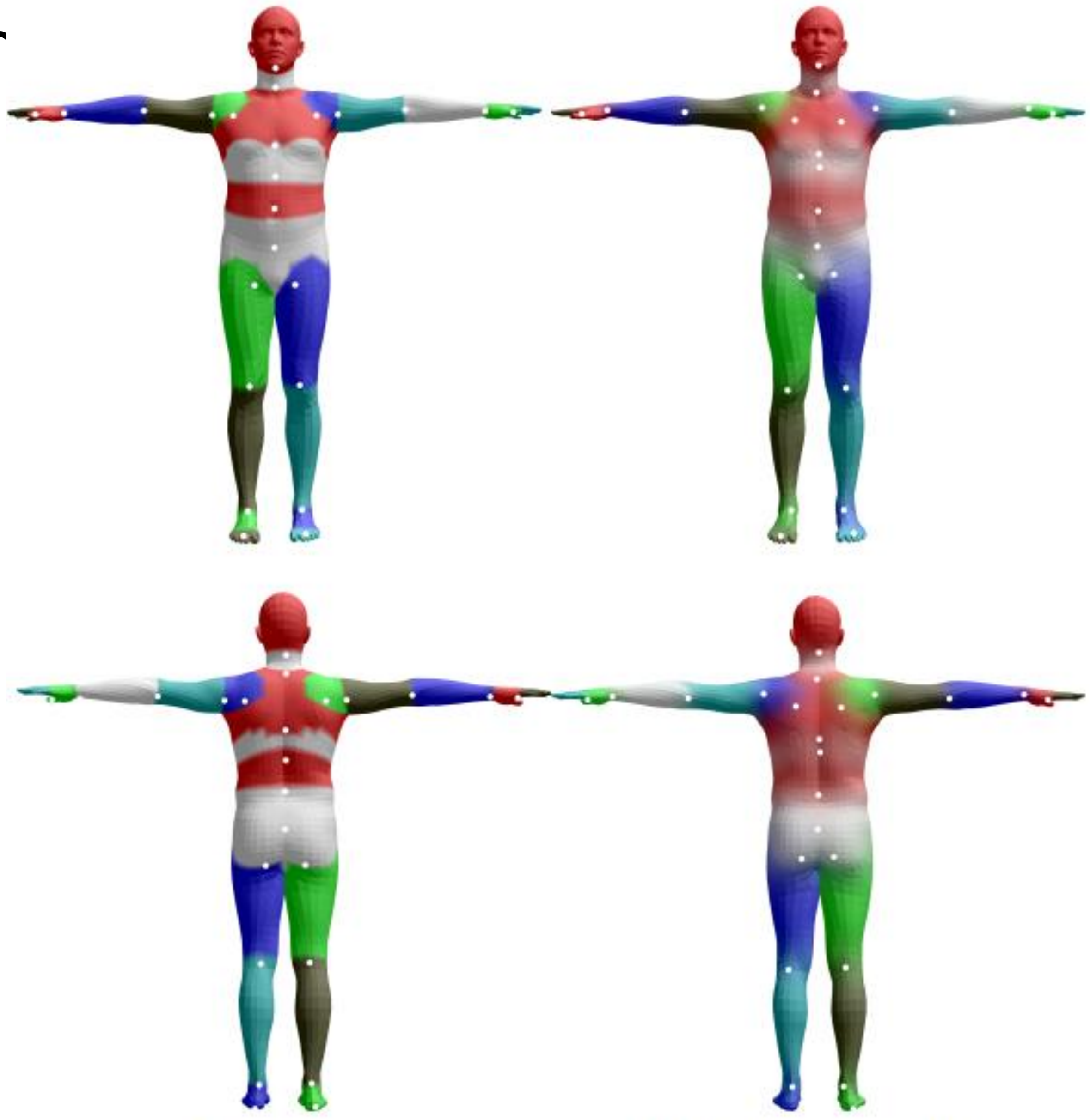
$$\lambda_U = 100$$

  - $U(\mathbf{T})$ finds a mirror image of vertices T, by flipping across the sagittal plane
  - Encourages symmetric template meshes and joint locations.

Initial estimate of the joir

- Model is hand segmented into 24 parts.
- $\mathcal{J}_I$ average ring of vertices connecting two parts.
- The estimating joints need to be close to the initial prediction

$$E_J(\hat{\mathbf{T}}^P, \hat{\mathbf{J}}^P) = \sum_{i=1}^{P_{\text{subj}}} ||\mathcal{J}_I \hat{\mathbf{T}}_i^P - \hat{\mathbf{J}}_i^P||^2$$



(a) *Segmentation* **(b)** *Initialization $\mathcal{W}_I$*

Regularization

- Pose-dependent blend shapes towards zero:

$$E_P(\mathcal{P}) = ||\mathcal{P}||_F^2$$

$$\|A\|_\mathrm{F} = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|^2}$$

- Blend weights towards the initial weights

$$E_W(\mathcal{W}) = ||\mathcal{W} - \mathcal{W}_I||_F^2.$$

The initial weights are computed by simply diffusing the segmentation.
-→ not simple for me

Training {W, P}

$$E_*(\hat{\mathbf{T}}^P, \hat{\mathbf{J}}^P, \Theta, \mathcal{W}, \mathcal{P}) =$$
$$E_D + \lambda_Y E_Y + \lambda_J E_J + \lambda_P E_P + E_W ,$$

where $\lambda_Y = 100$, $\lambda_J = 100$ and $\lambda_P = 25$.

Joint regressor

Body mesh

↓

Non –negative least square [Lawson and Hanson 1995]???

↓

regressor matrix $\mathcal{J}$

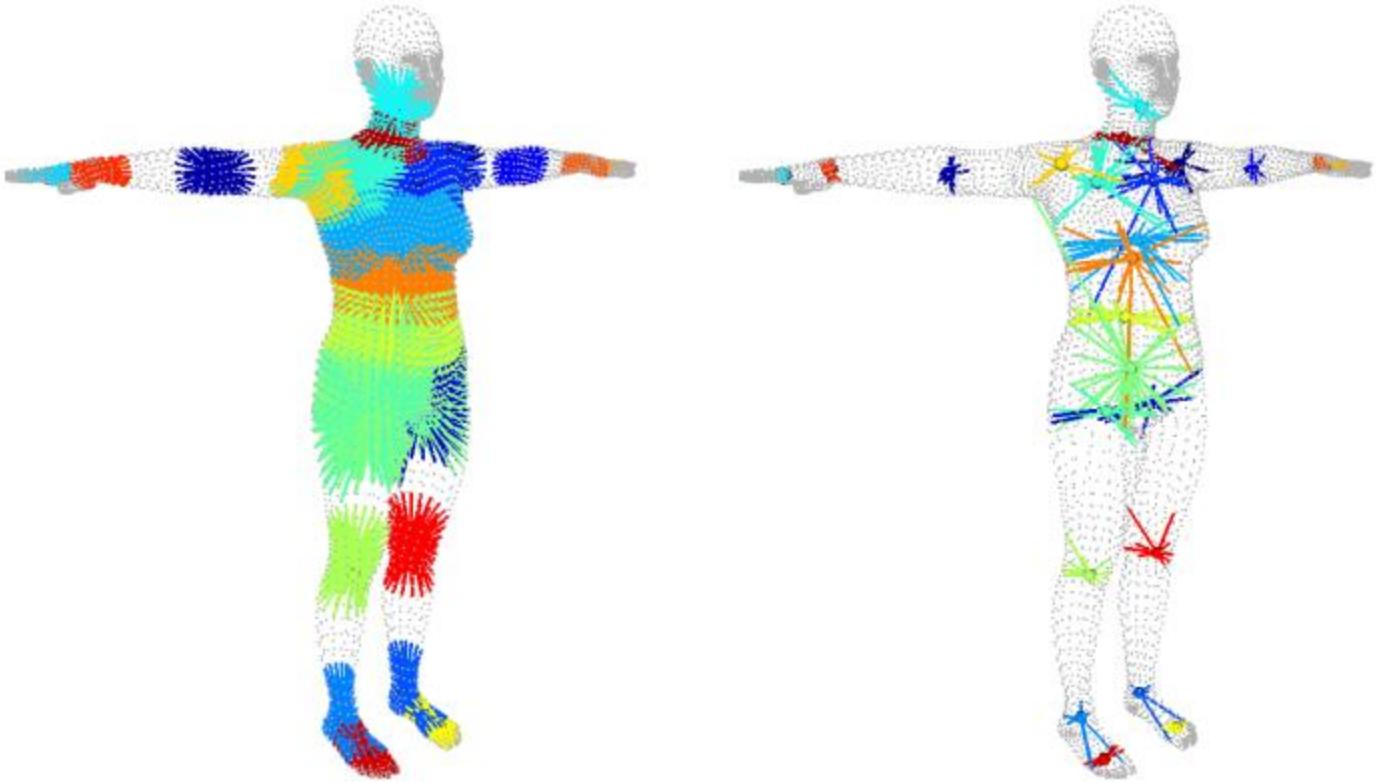
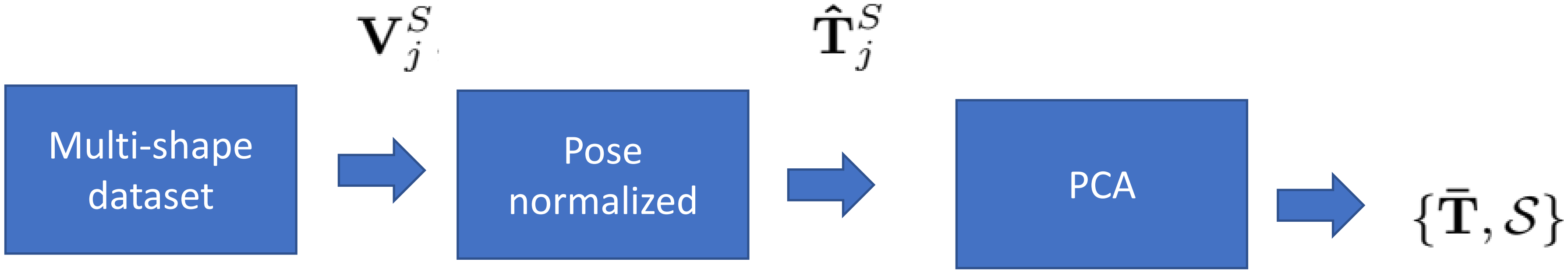


Figure 7: Joint regression. *(left) Initialization. Joint locations can be influenced by locations on the surface, indicated by the colored lines. We assume that these influences are somewhat local. (right) Optimized. After optimization we find a sparse set of vertices and associated weights influencing each joint.*

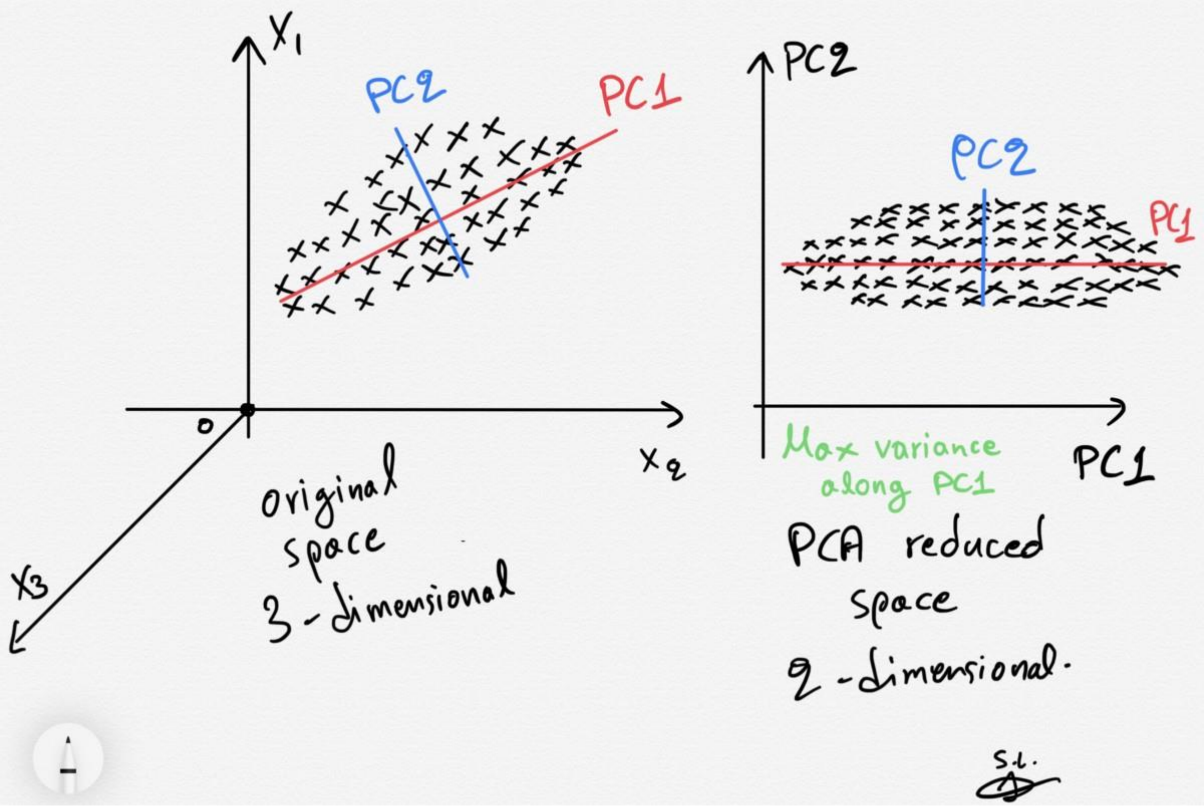
Shape parameter training

- $\{\bar{\mathbf{T}}, \mathcal{S}\}$: mean and principal shape directions

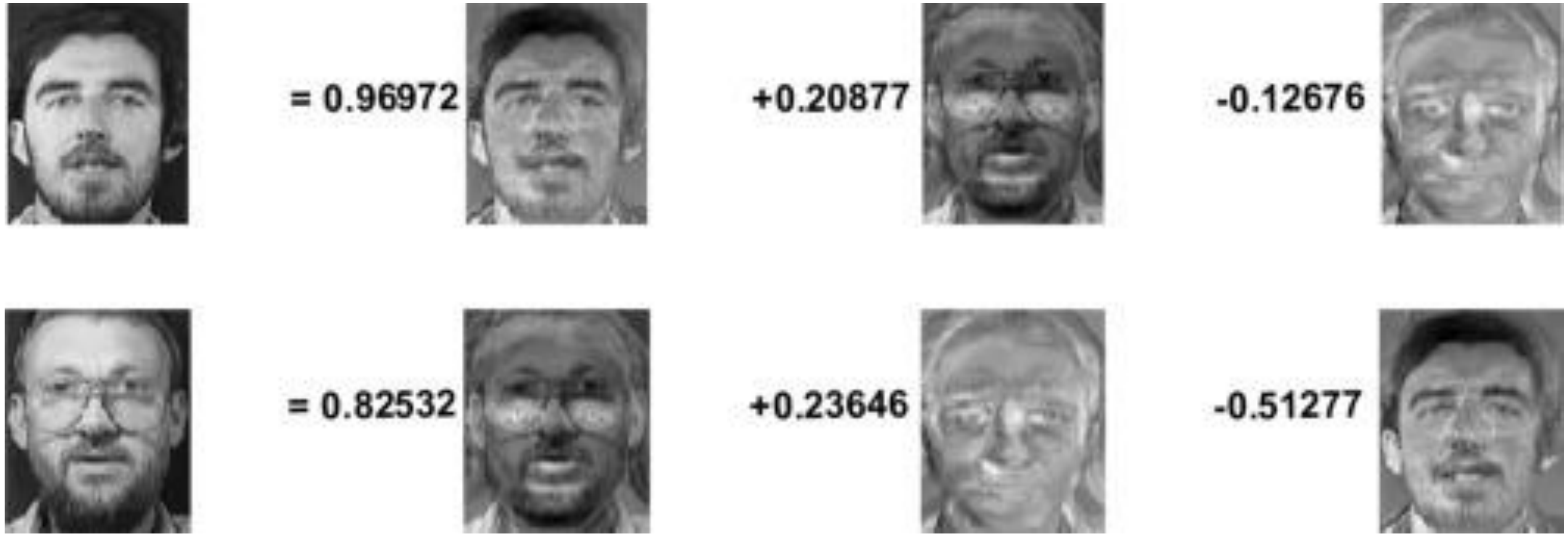


Principal component analysis (PCA)

- Reduce dimensionality



Application to face



$\hat{\mathbf{T}}_{\mu}^{P}$ and $\hat{\mathbf{J}}_{\mu}^{P}$ From multi-pose dataset

mean shape Mean joint locations

- An edge of the model and the registration $W_e(\hat{\mathbf{T}}_{\mu}^{P}, \hat{\mathbf{J}}_{\mu}^{P}, \vec{\theta}, \mathcal{W}), \mathbf{V}_{j,e}^{S} \in \mathbb{R}^3$
-

$\mathbf{V}_j^S$ → Estimate pose blend shape →

Raw registration mesh

$\{\mathcal{J}, \mathcal{W}, \mathcal{P}\}$

From multi-pose dataset

Find pose

$$\vec{\theta}_j = \underset{\vec{\theta}}{\arg\min} \sum_e ||W_e(\hat{\mathbf{T}}_{\mu}^{P} + B_P(\vec{\theta}; \mathcal{P}), \hat{\mathbf{J}}_{\mu}^{P}, \vec{\theta}, \mathcal{W}) - \mathbf{V}_{j,e}^{S}||^2$$

↓

Find pose-normalized template

$$\hat{\mathbf{T}}_j^S = \underset{\hat{\mathbf{T}}}{\arg\min} ||W(\hat{\mathbf{T}} + B_P(\vec{\theta}_j; \mathcal{P}), \mathcal{J}\hat{\mathbf{T}}, \vec{\theta}_j, \mathcal{W}) - \mathbf{V}_j^S||^2$$

Evaluation