Virtual Humans – Winter 23/24

Lecture 5_1 – Training a Body Model and Fitting SMPL to Scans

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

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Agenda

- Body models based on triangle deformations (SCAPE, BlendSCAPE) – very briefly
- Training a Body Model from registrations
- How to obtain registrations (Fitting SMPL)
- Jointly solve for registration and model training

Two deformation models

- Local triangle deformations
- 3x3 transformations
- Applied to vertex differences (edges)
- -> BlendSCAPE

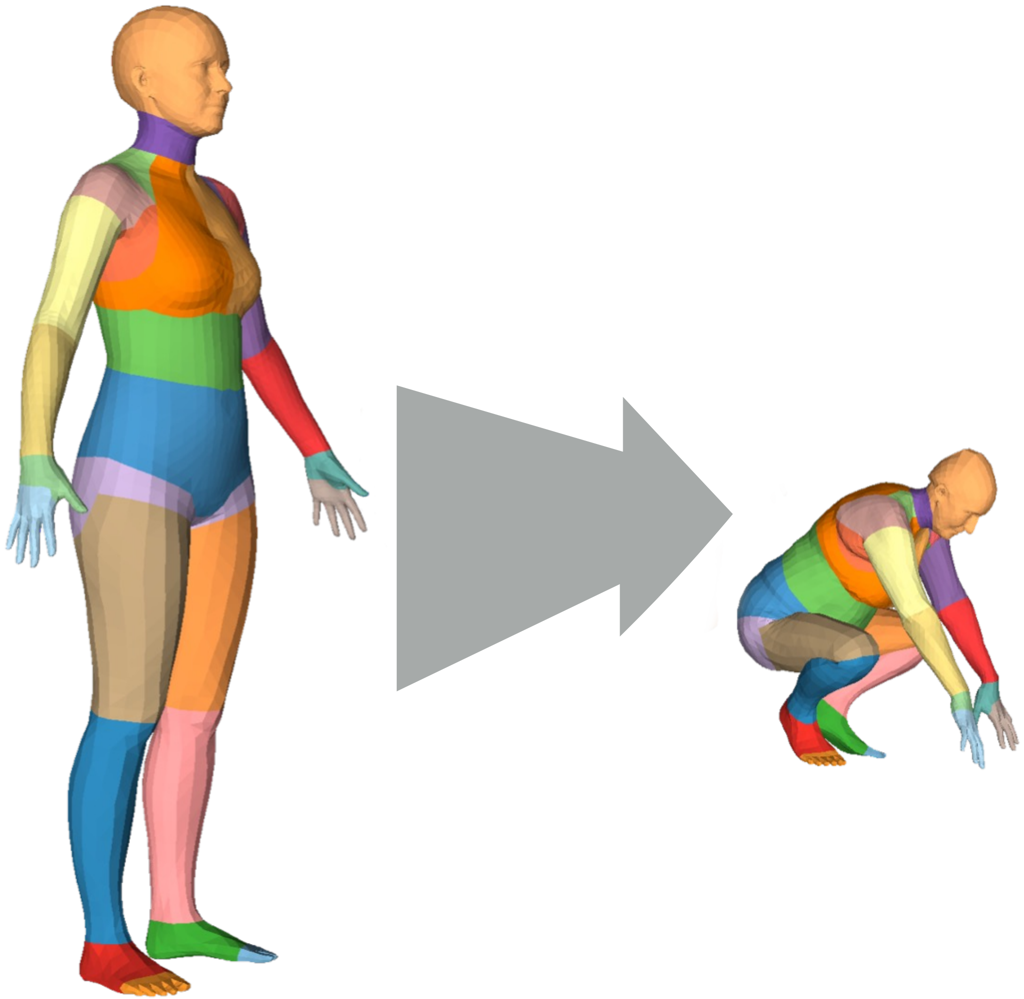
- Global vertex deformations
- 4x4 transformations
- Applied to vertices
- -> SMPL

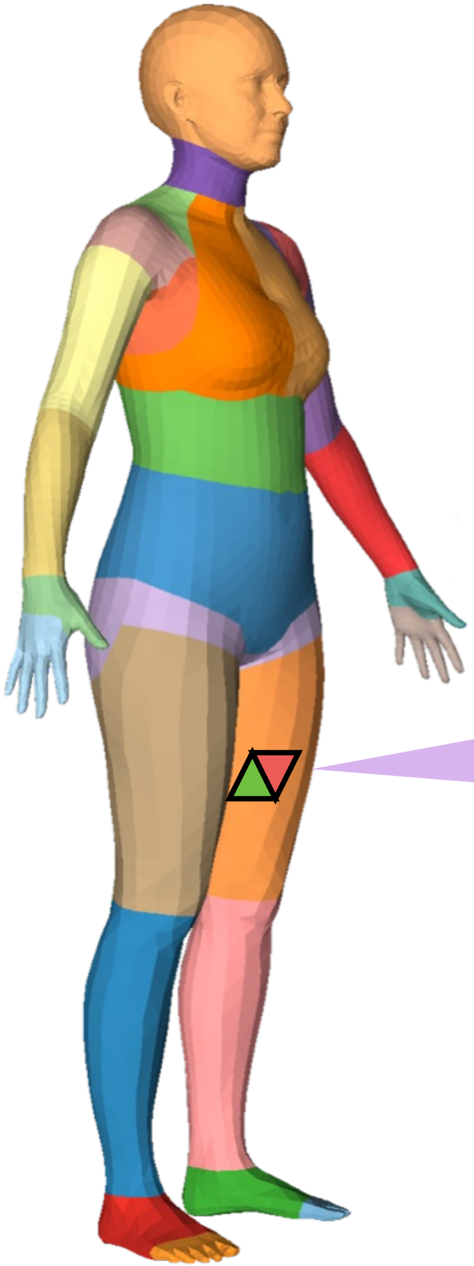
Two deformation models

- Local triangle deformations
- 3x3 transformations
- Applied to vertex differences (edges)
- -> BlendSCAPE

- Global vertex deformations
- 4x4 transformations
- Applied to vertices
- -> SMPL

Modelling with local triangle deformations: SCAPE and BlendSCAPE

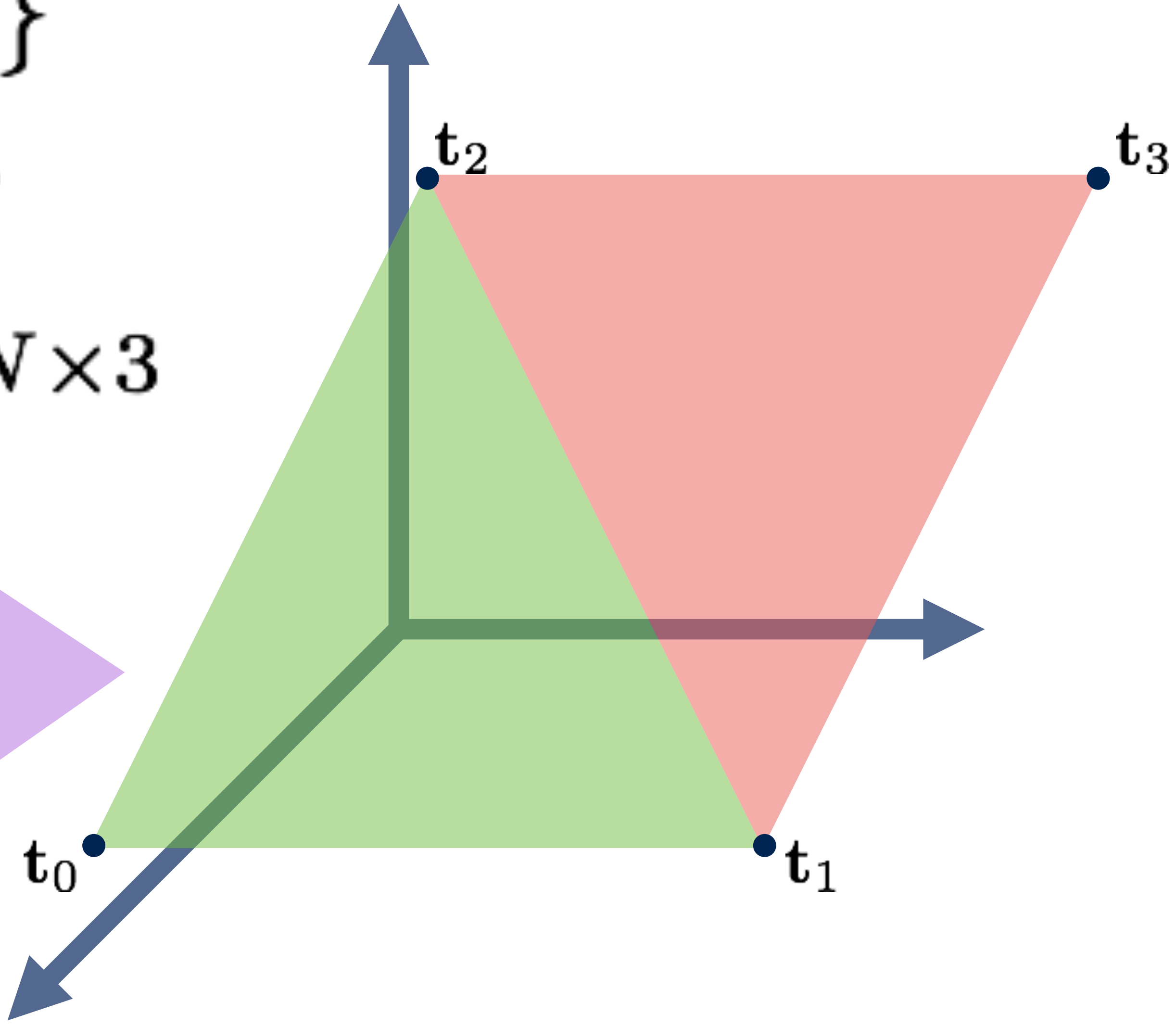


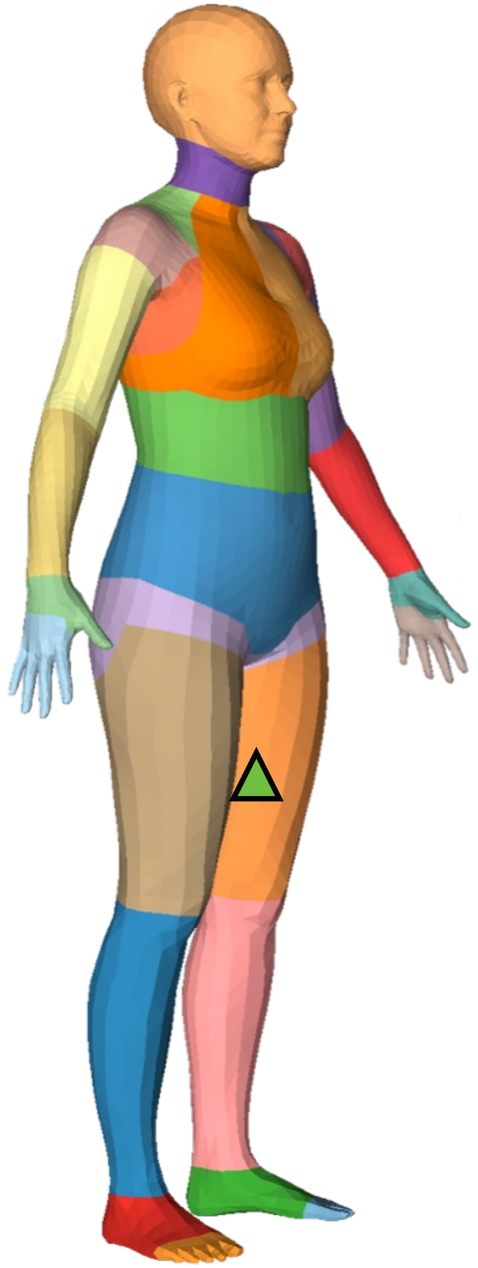


$$\bar{\mathbf{T}} = \{\mathbf{t}\}$$

$$\mathbf{t}_i \in \mathbb{R}^3$$

$$\bar{\mathbf{T}} \in \mathbb{R}^{N \times 3}$$



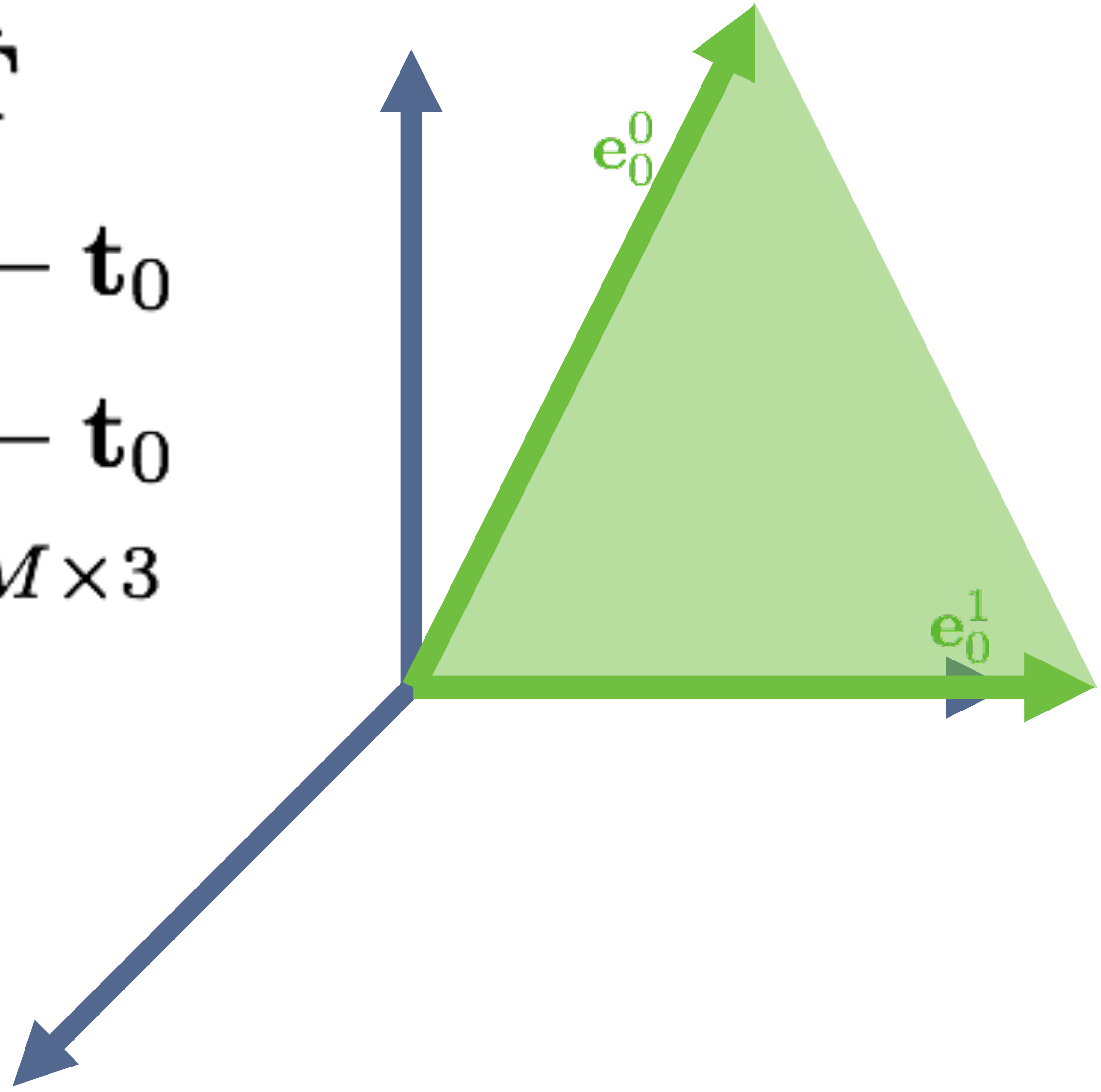


$$\mathbf{E} = A\bar{\mathbf{T}}$$

$$\mathbf{e}_0^0 = \mathbf{t}_2 - \mathbf{t}_0$$

$$\mathbf{e}_0^1 = \mathbf{t}_1 - \mathbf{t}_0$$

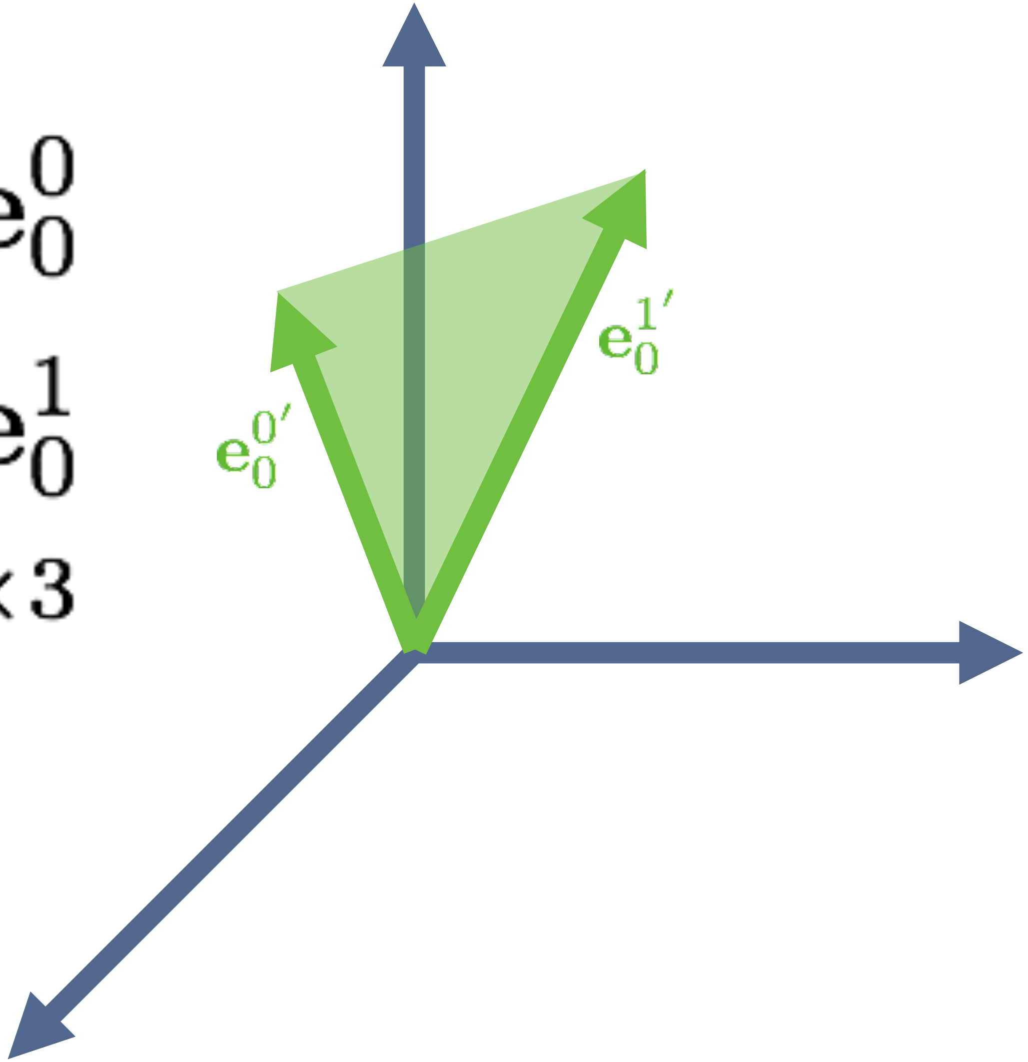
$$\mathbf{E} \in \mathbb{R}^{2M \times 3}$$

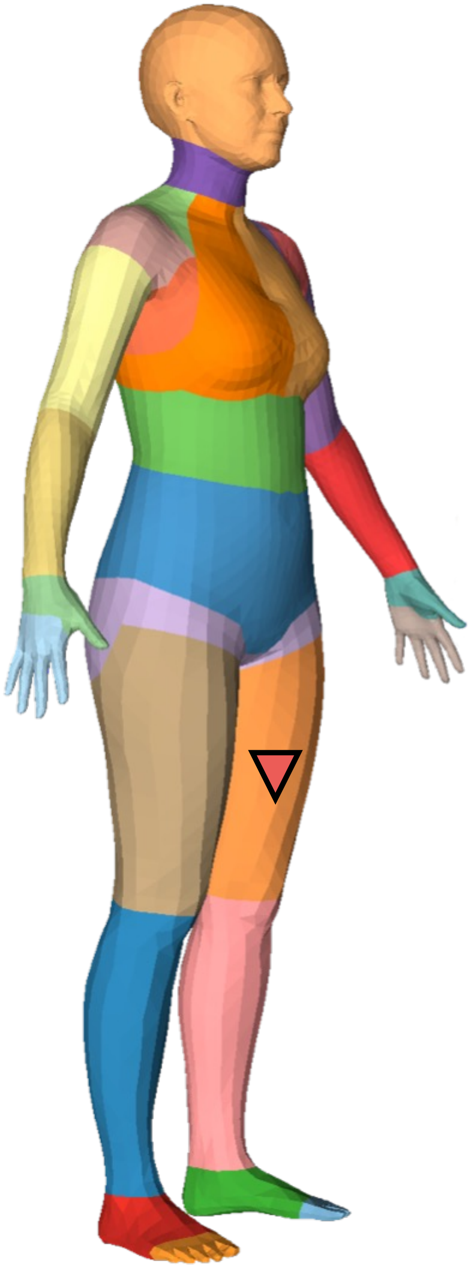


$$\mathbf{e}_0^{0'} = X_0 \mathbf{e}_0^0$$

$$\mathbf{e}_0^{1'} = X_0 \mathbf{e}_0^1$$

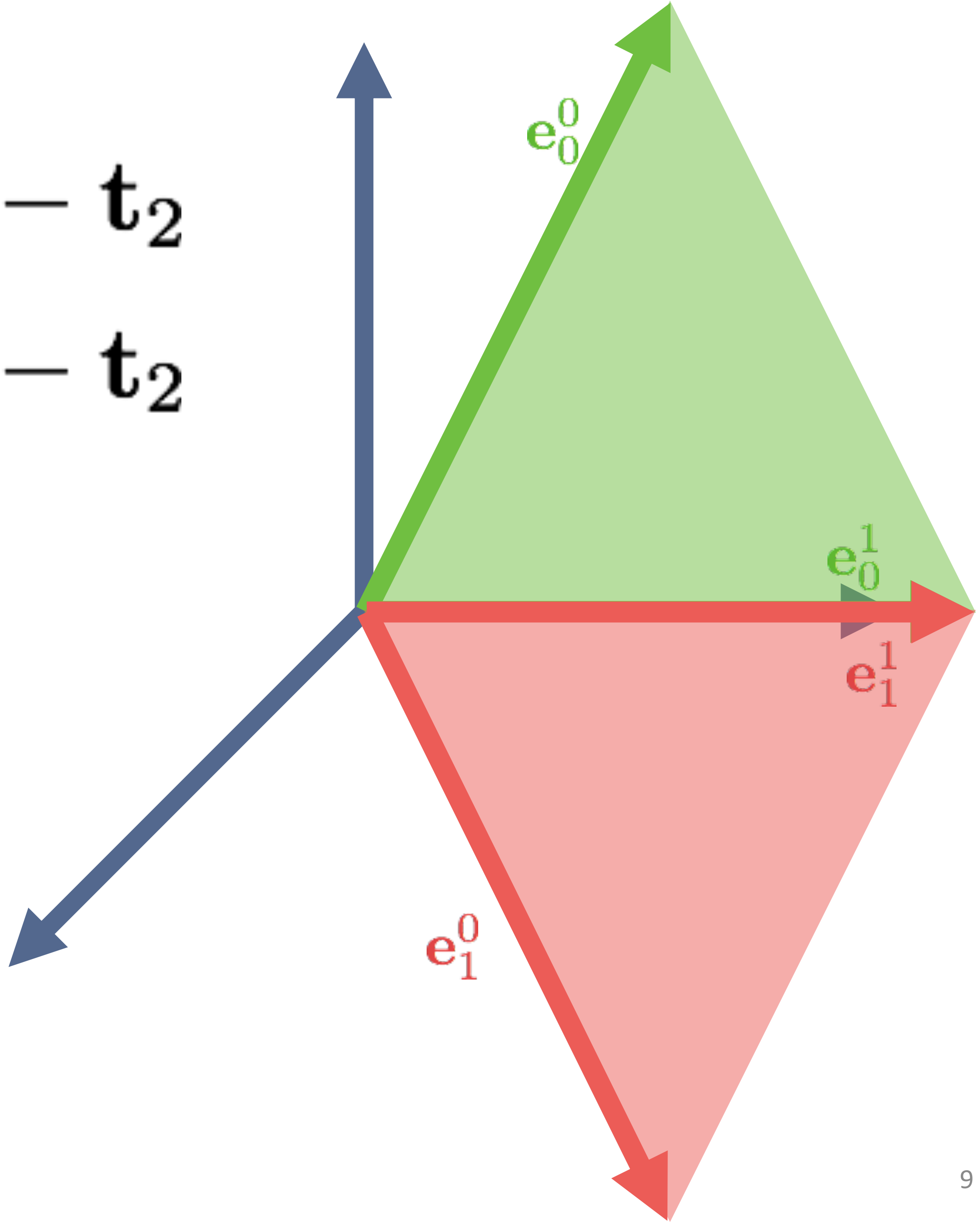
$$X_0 \in \mathbb{R}^{3\times 3}$$





$$\mathbf{e}_1^0 = \mathbf{t}_1 - \mathbf{t}_2$$

$$\mathbf{e}_1^1 = \mathbf{t}_3 - \mathbf{t}_2$$



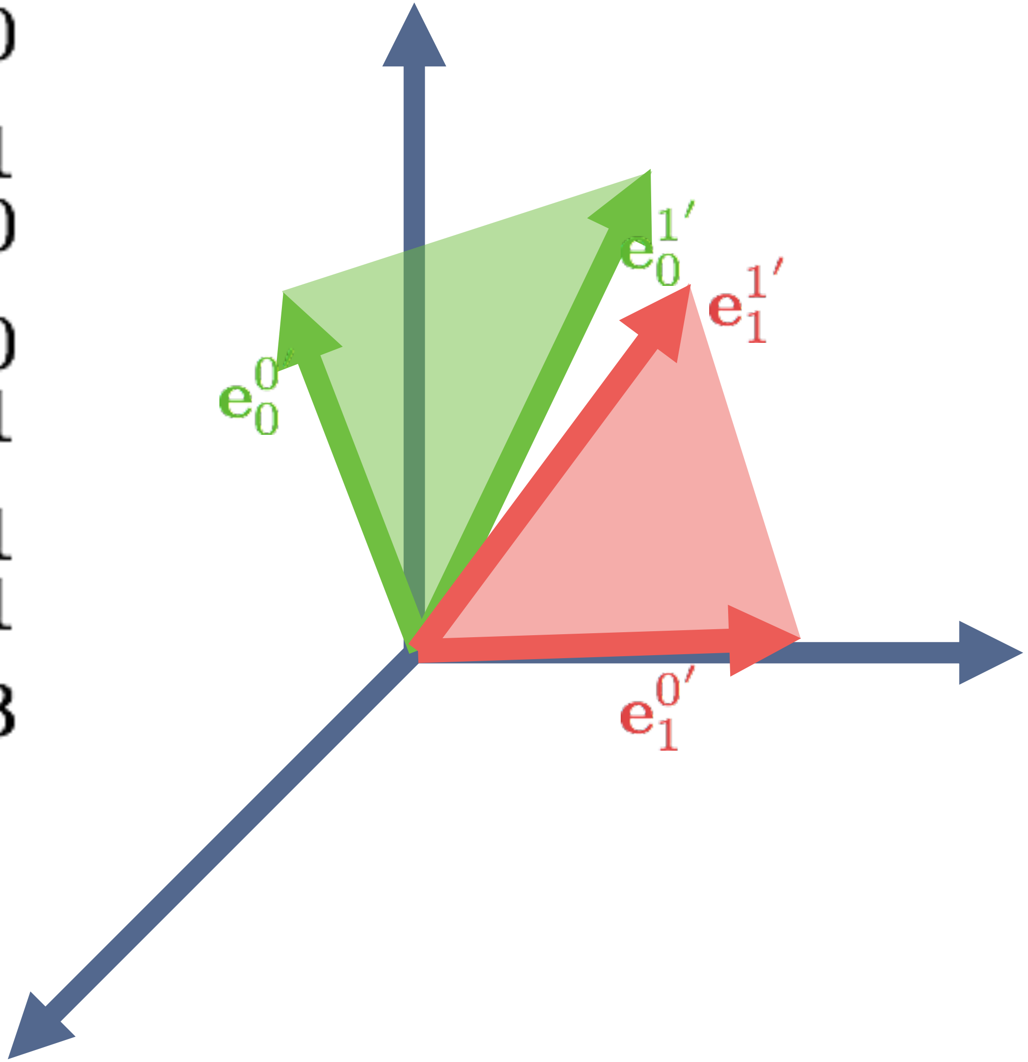
$$\mathbf{e}_0^{0'} = X_0\mathbf{e}_0^0$$

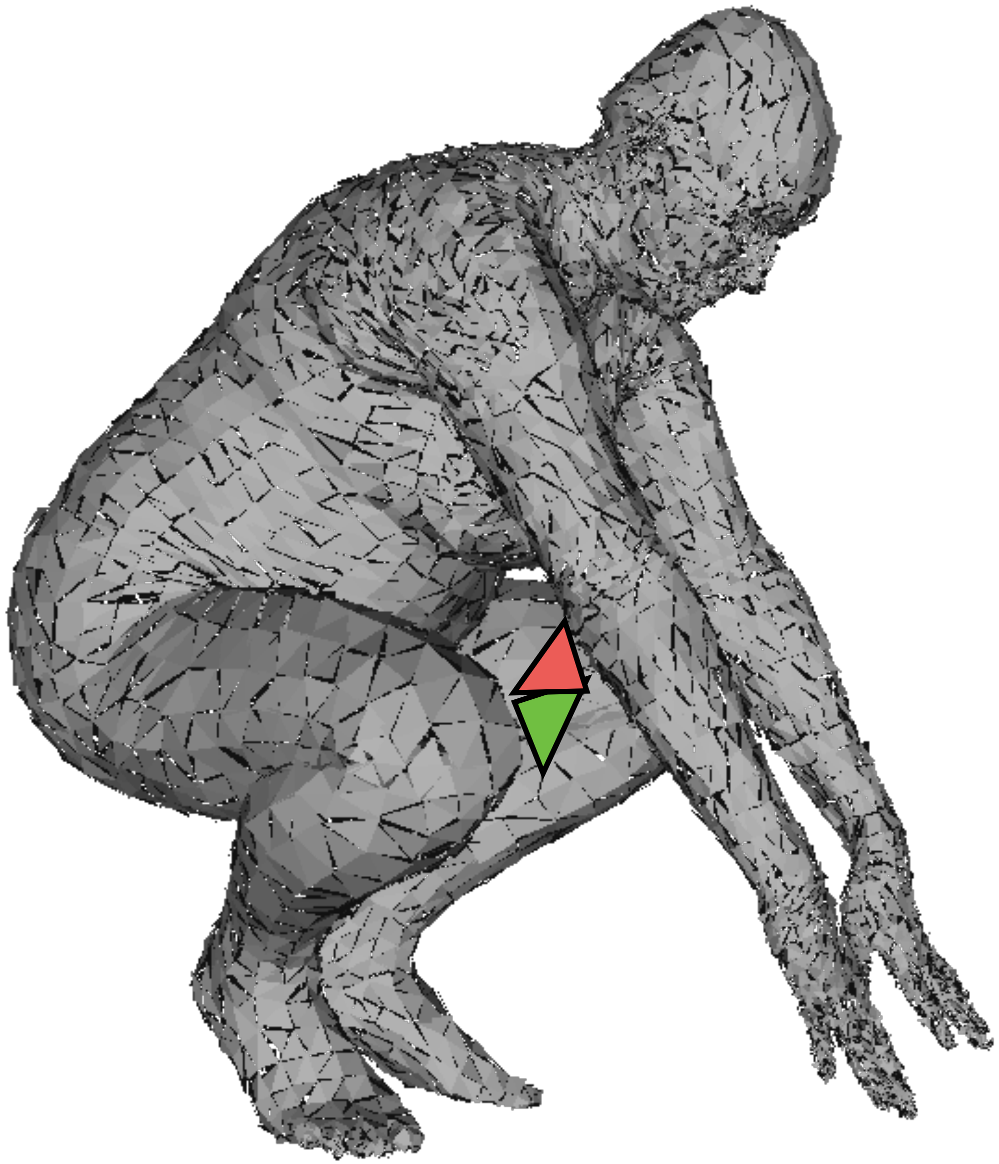
$$\mathbf{e}_0^{1'} = X_0\mathbf{e}_0^1$$

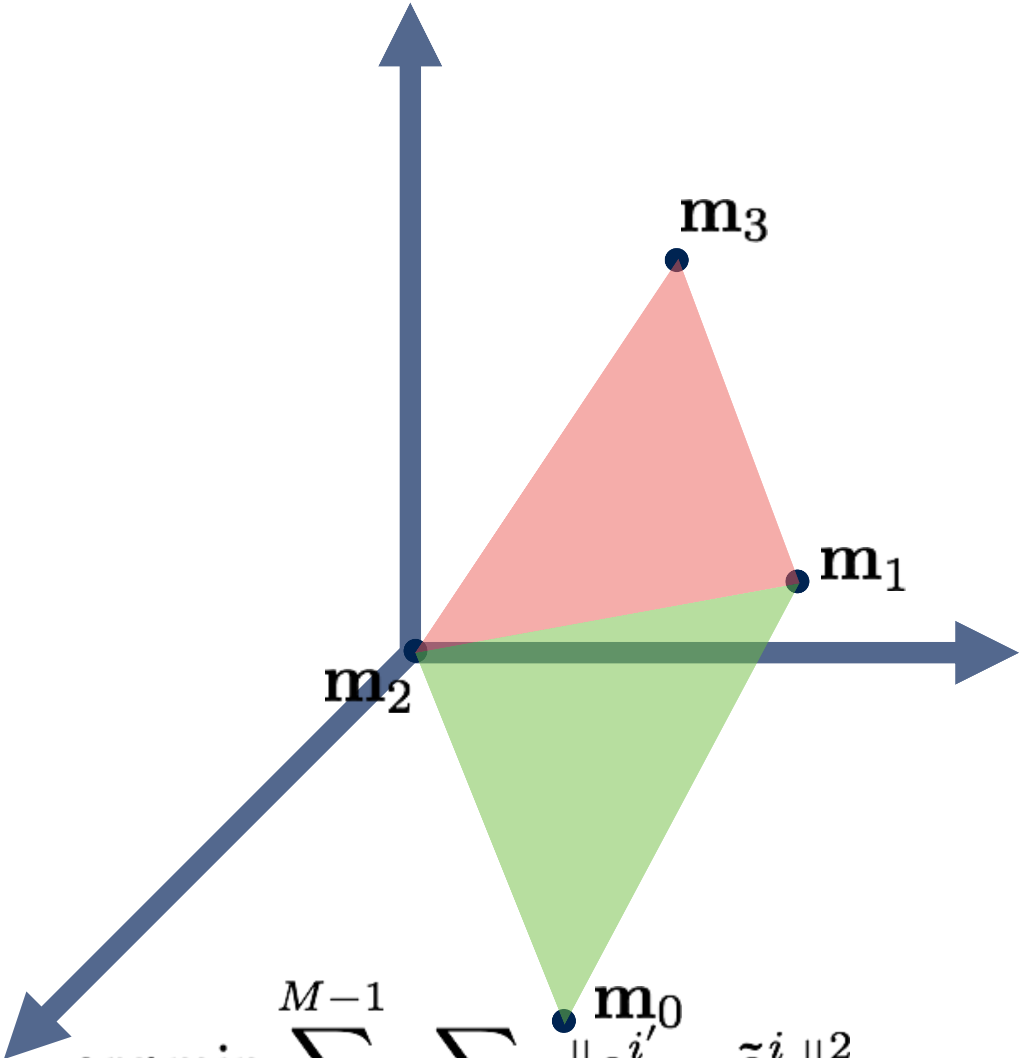
$$\mathbf{e}_1^{0'} = X_1\mathbf{e}_1^0$$

$$\mathbf{e}_1^{1'} = X_1\mathbf{e}_1^1$$

$$X_f \in \mathbb{R}^{3\times 3}$$

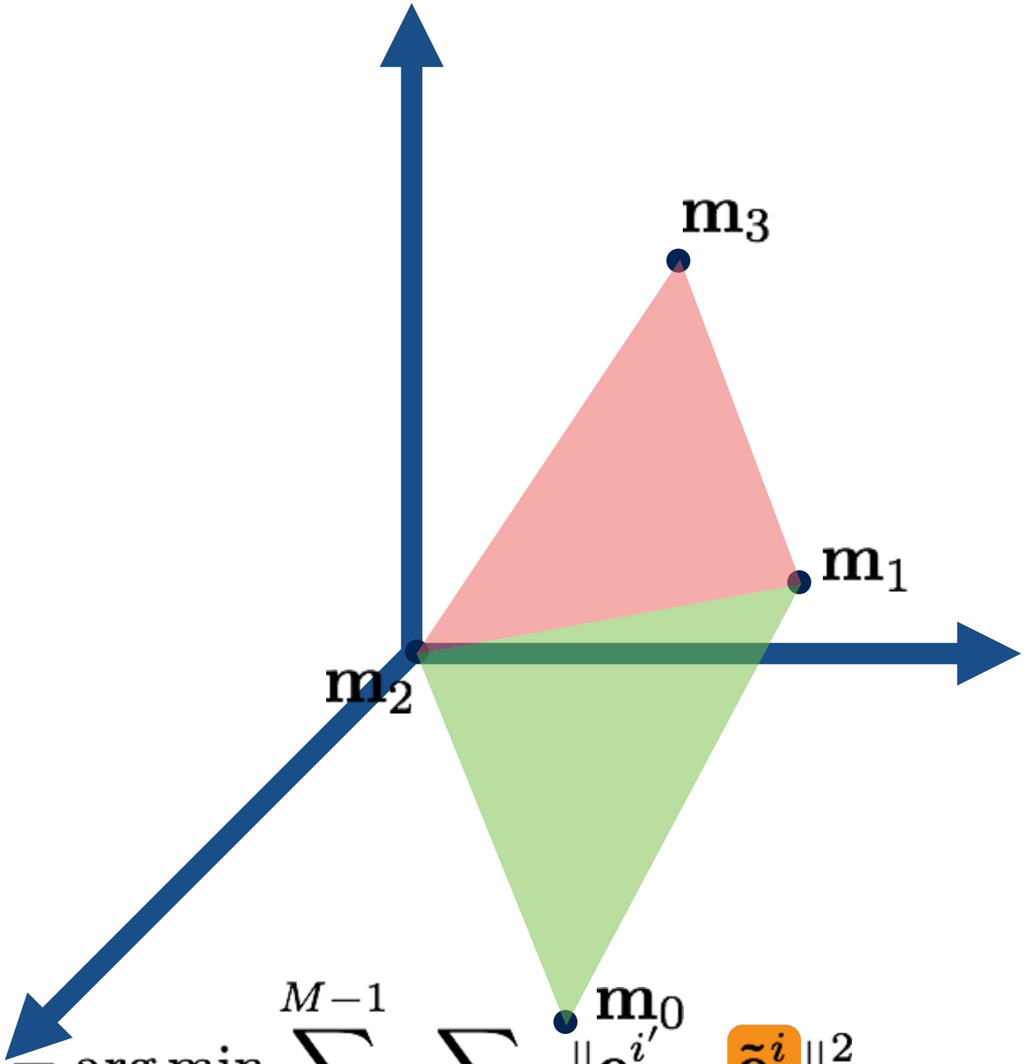






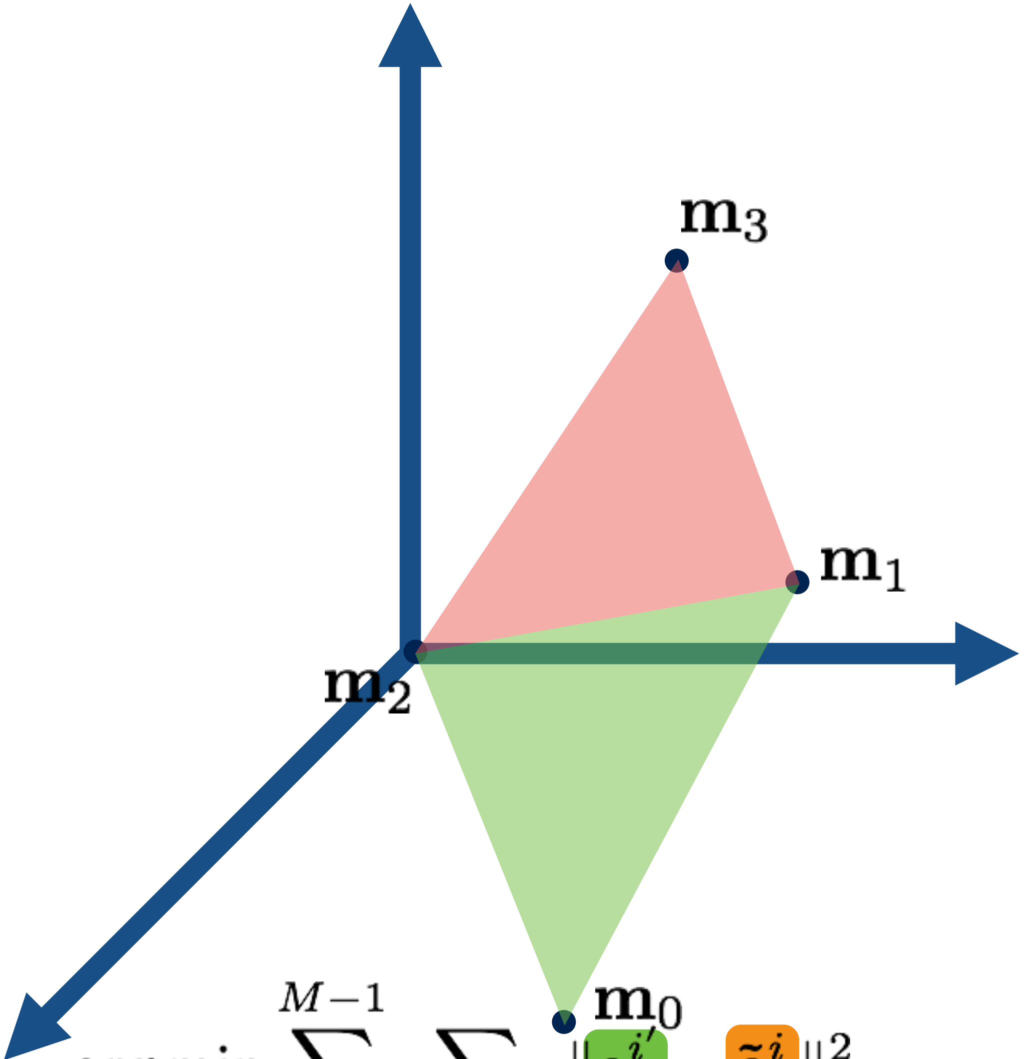
$$\mathbf{M} = \arg\min_{\mathbf{M}} \sum_{f=0}^{M-1} \sum_{i\in\{0,1\}} \|\mathbf{e}_f^{i'} - \tilde{\mathbf{e}}_f^i\|^2$$

$$\tilde{\mathbf{E}} \equiv A\mathbf{M} = \{\tilde{\mathbf{e}}\}$$



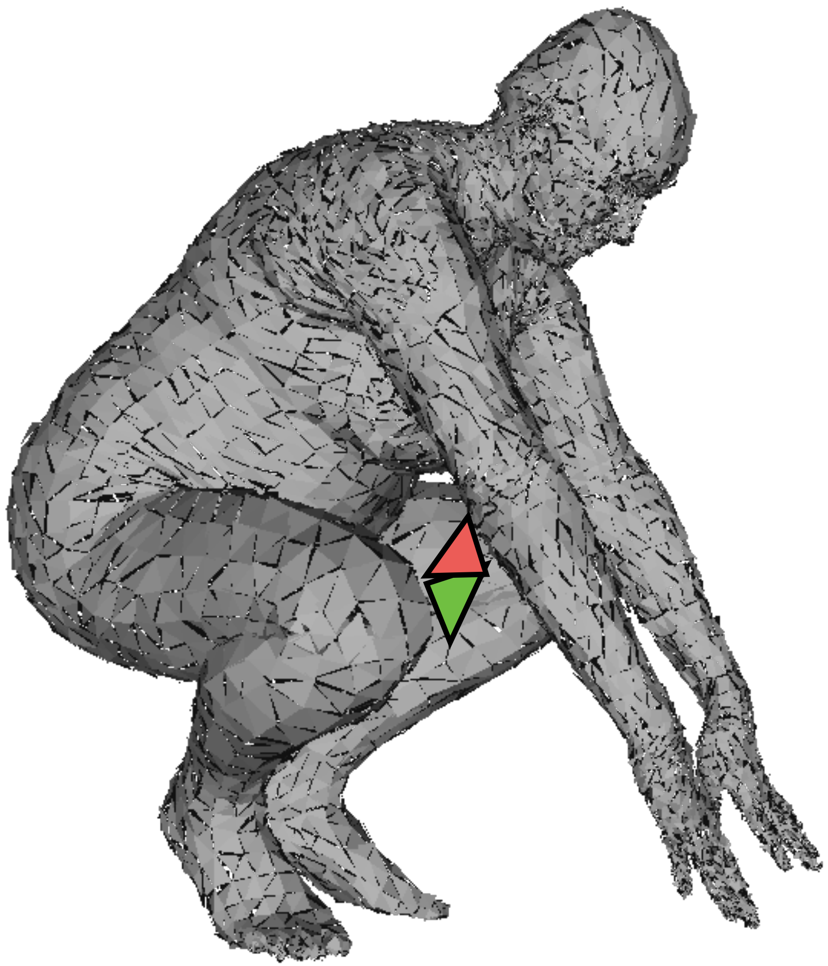
$$\mathbf{M} = \arg\min_{\mathbf{M}} \sum_{f=0}^{M-1} \sum_{i\in\{0,1\}} \|\mathbf{e}_f^{i'} - \tilde{\mathbf{e}}_f^{i}\|^2$$

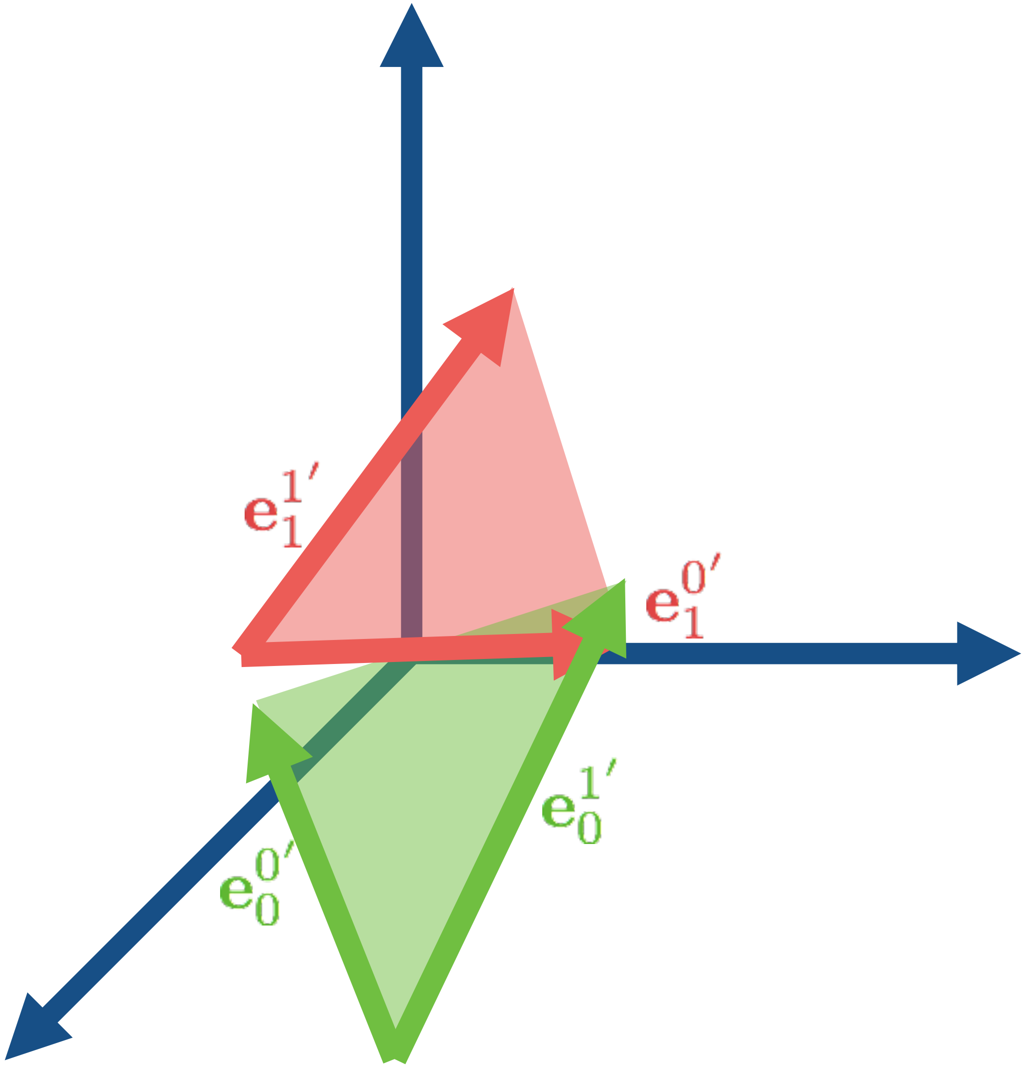
$$\tilde{\mathbf{E}} \equiv A\mathbf{M} = \{\tilde{\mathbf{e}}\}$$



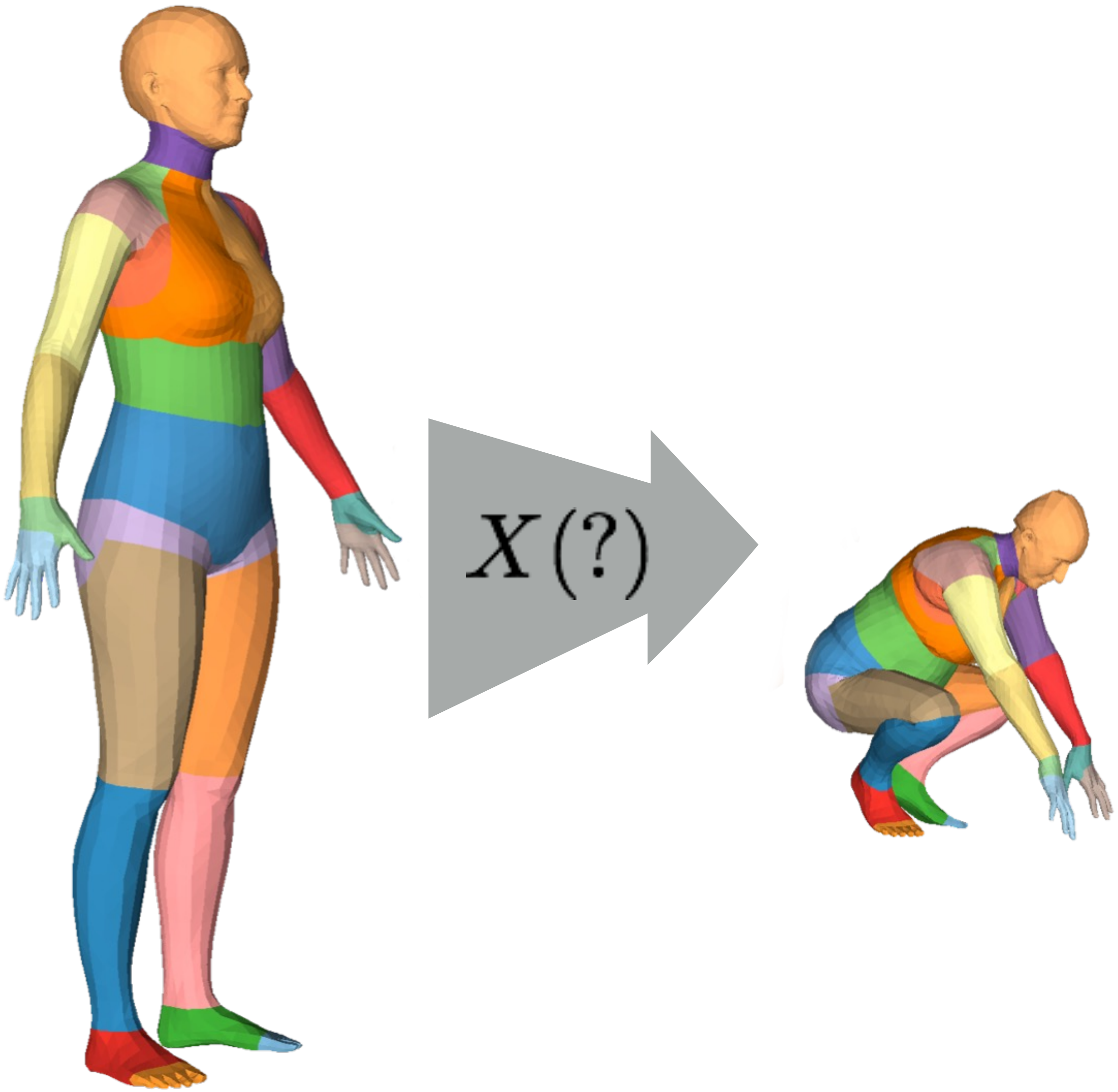
$$\mathbf{M} = \arg\min_{\mathbf{M}} \sum_{f=0}^{M-1} \sum_{i\in\{0,1\}} \|\mathbf{e}_f^{i'} - \tilde{\mathbf{e}}_f^{i}\|^2$$

$$\tilde{\mathbf{E}} \equiv A\mathbf{M} = \{\tilde{\mathbf{e}}\}$$

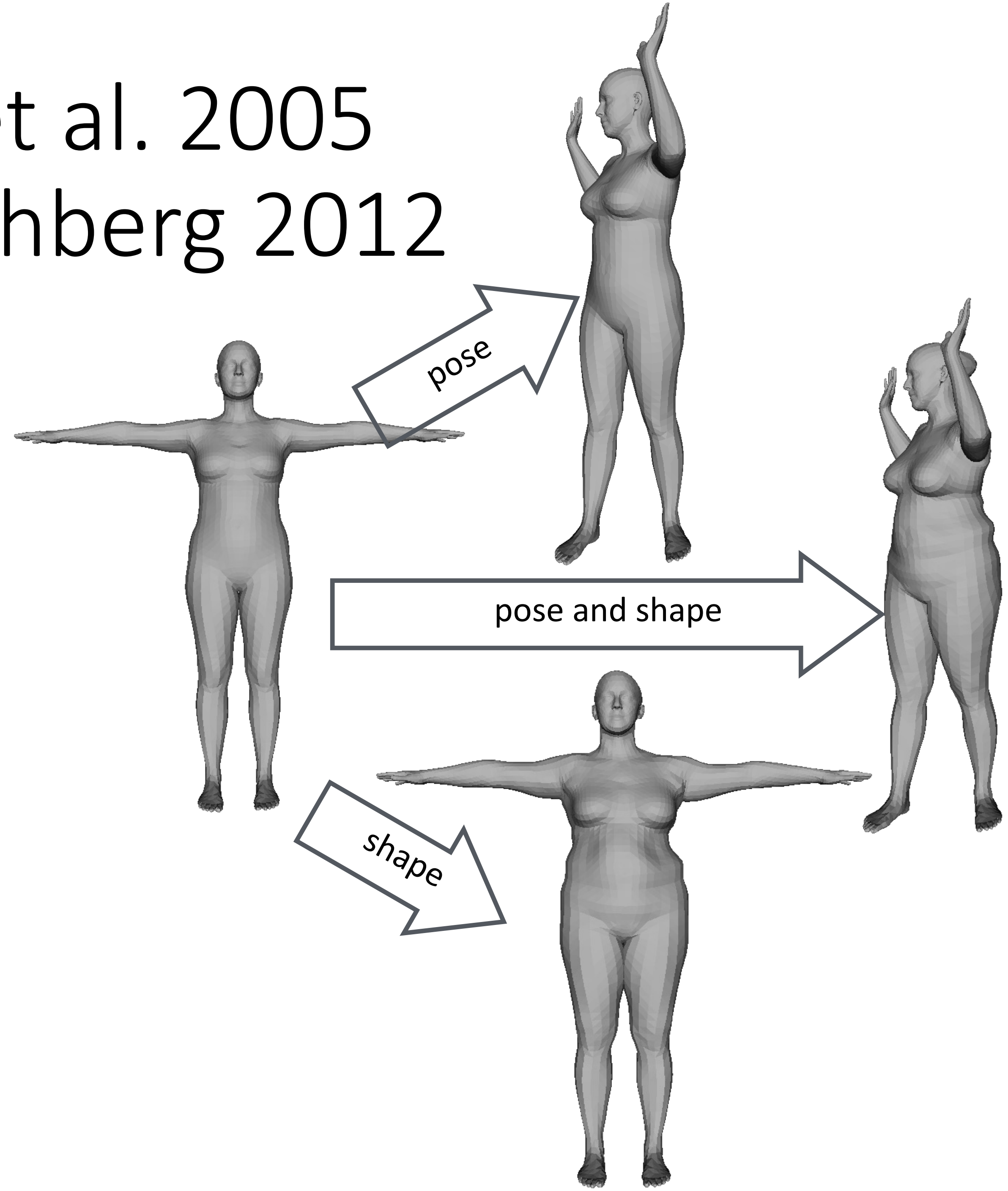



$e_1^{1'}$
$e_1^{0'}$
$e_0^{1'}$
$e_0^{0'}$

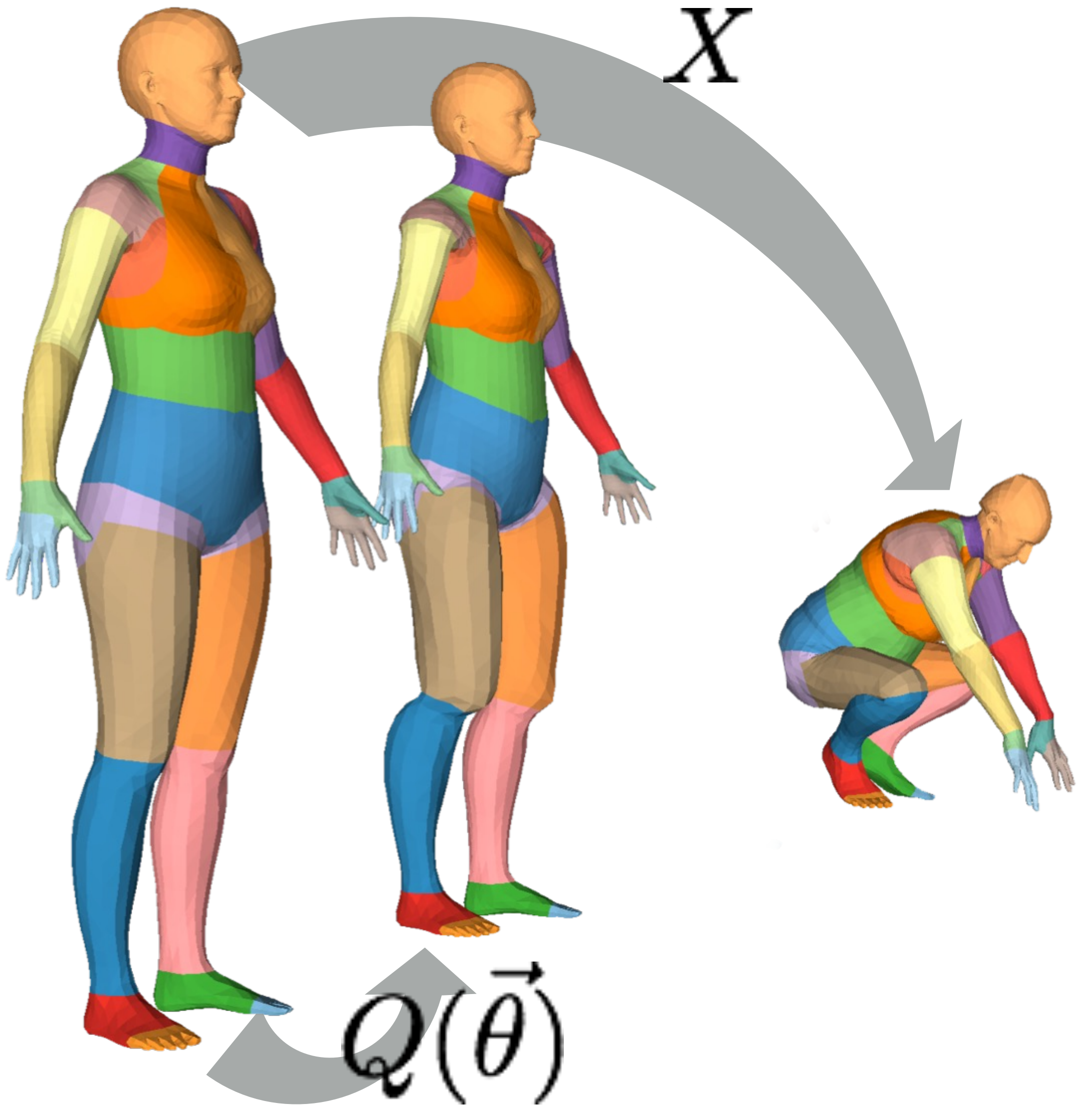
Modeling deformations



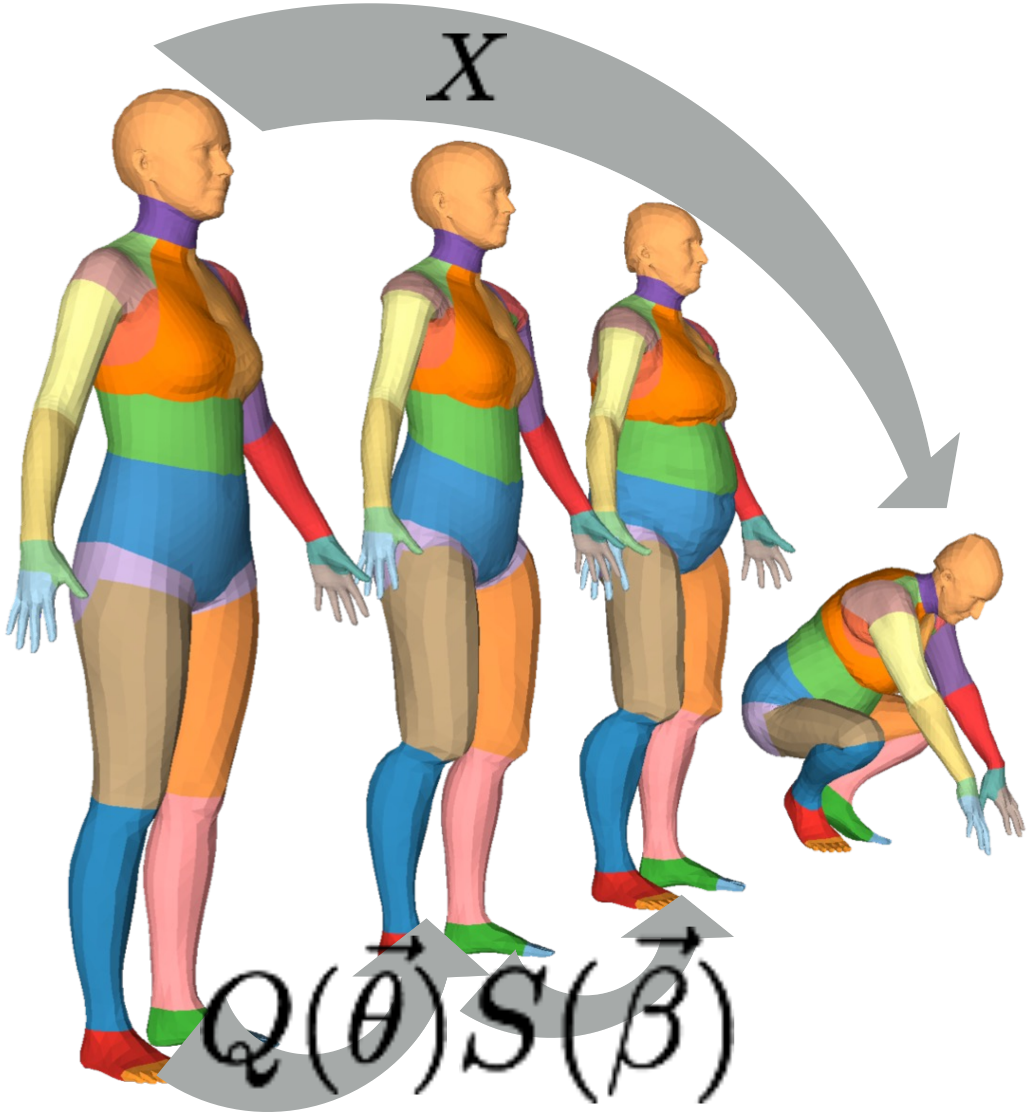
SCAPE. Angelov et al. 2005
BlendSCAPE, Hirshberg 2012



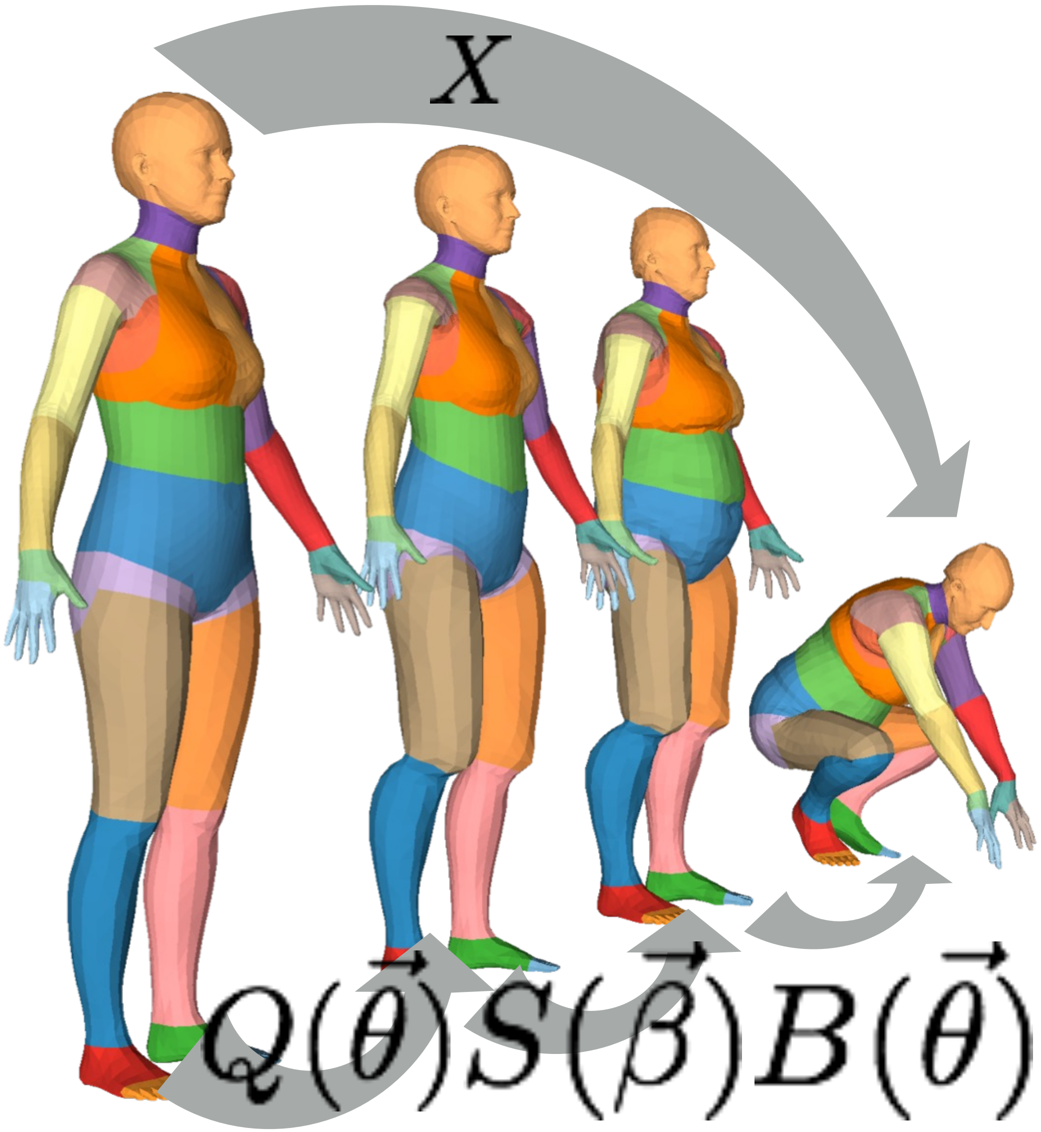
SCAPE and BlendSCAPE



SCAPE and BlendSCAPE



SCAPE and BlendSCAPE



SMPL Additive Model

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

Joint locations

Vertices

BlendSCAPE Multiplicative Model

$$\mathbf{e}' = B(\vec{\theta})S(\vec{\beta})Q(\vec{\theta})\mathbf{e}$$

No joints

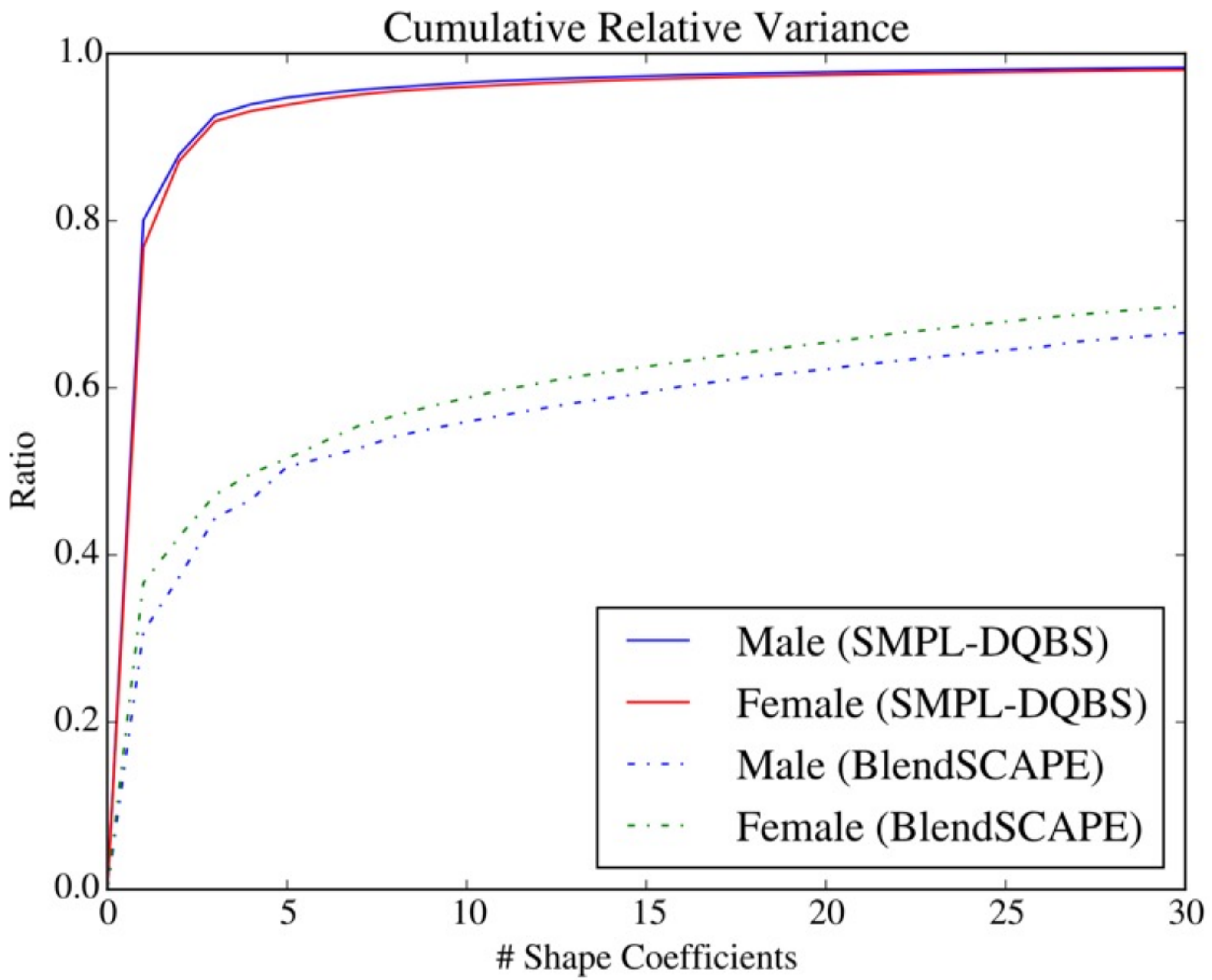
Edges

Two deformation models

- Local triangle deformations
- 3x3 transformations
- Applied to vertex differences (edges)
- -> BlendSCAPE

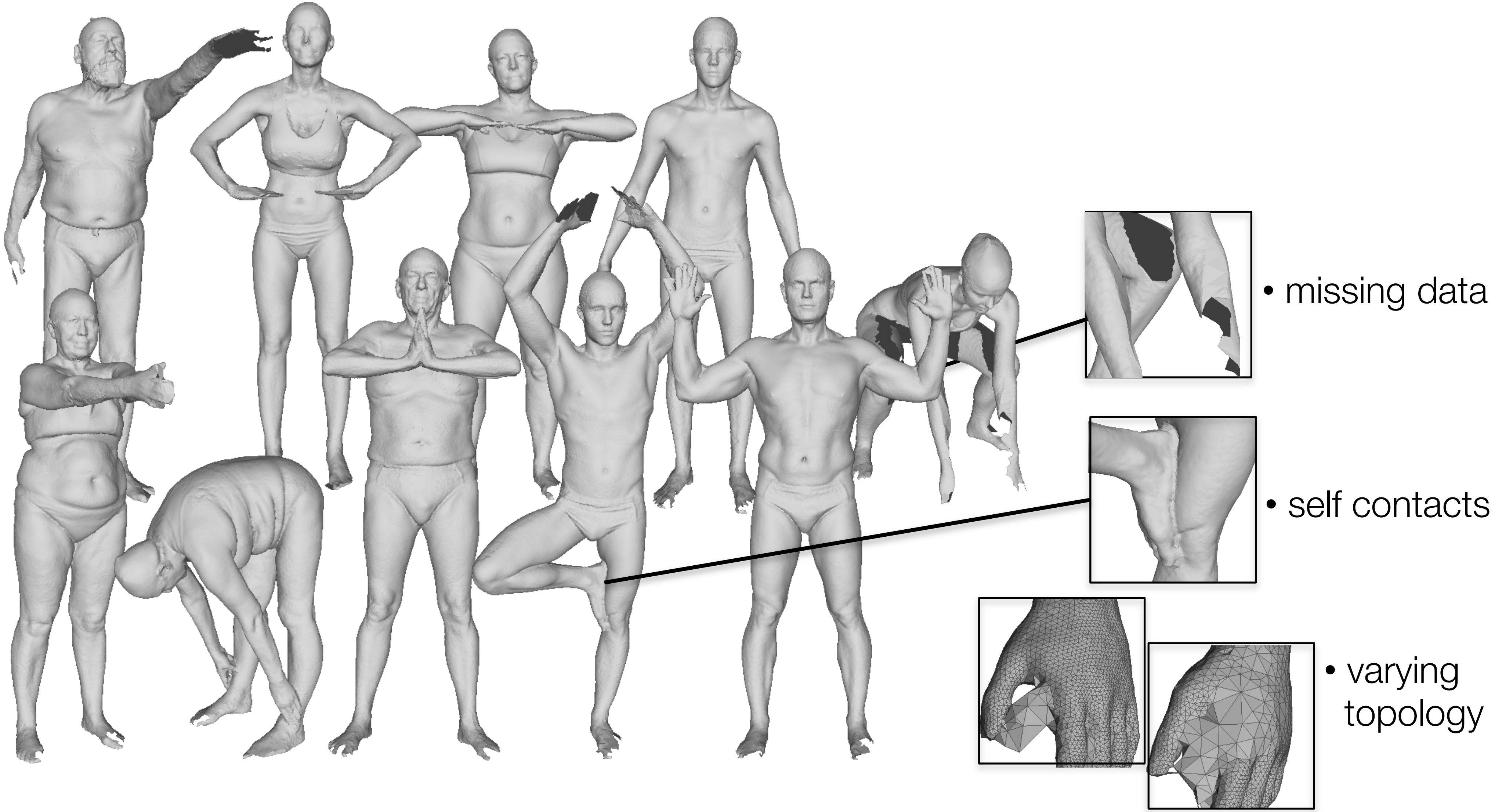
- Global vertex deformations
- 4x4 transformations
- Applied to vertices
- -> SMPL

SMPL vs SCAPE



How do we train a body model ?

Scans



Training

$$\Phi = \arg\min_{\Phi} \sum_{j} \min_{\vec{\theta_j}, \vec{\beta_j}} \| M(\vec{\theta_j}, \vec{\beta_j}; \Phi) - \mathbf{V}_j \|^2$$

Training

$$\Phi = \arg\min_{\Phi} \sum_j \min_{\vec{\theta}_j, \vec{\beta}_j} \| M(\vec{\theta}_j, \vec{\beta}_j; \Phi) - \mathbf{V}_j \|^2$$

Parameters to be learned

Training

$$\Phi = \arg\min_{\Phi} \sum_j \min_{\vec{\theta}_j, \vec{\beta}_j} \| M(\vec{\theta}_j, \vec{\beta}_j; \Phi) - \mathbf{V}_j \|^2$$

Parameters to be learned

Model

Training

$$\Phi = \arg\min_{\Phi} \sum_j \min_{\vec{\theta}_j, \vec{\beta}_j} \| M(\vec{\theta}_j, \vec{\beta}_j; \Phi) - \mathbf{V}_j \|^2$$

Parameters to be learned

Model

Set of registrations

Training

$$\Phi = \arg\min_{\Phi} \sum_j \min_{\vec{\theta}_j, \vec{\beta}_j} \| M(\vec{\theta}_j, \vec{\beta}_j; \Phi) - \mathbf{V}_j \|^2$$

Parameters to be learned

Model

Set of registrations

Shape parameters of registration j

Training

$$\Phi = \arg\min_{\Phi} \sum_j \min_{\vec{\theta}_j, \vec{\beta}_j} \| M(\vec{\theta}_j, \vec{\beta}_j; \Phi) - \mathbf{V}_j \|^2$$

Parameters to be learned

Model

Set of registrations

Shape parameters of registration j

Pose parameters of registration j

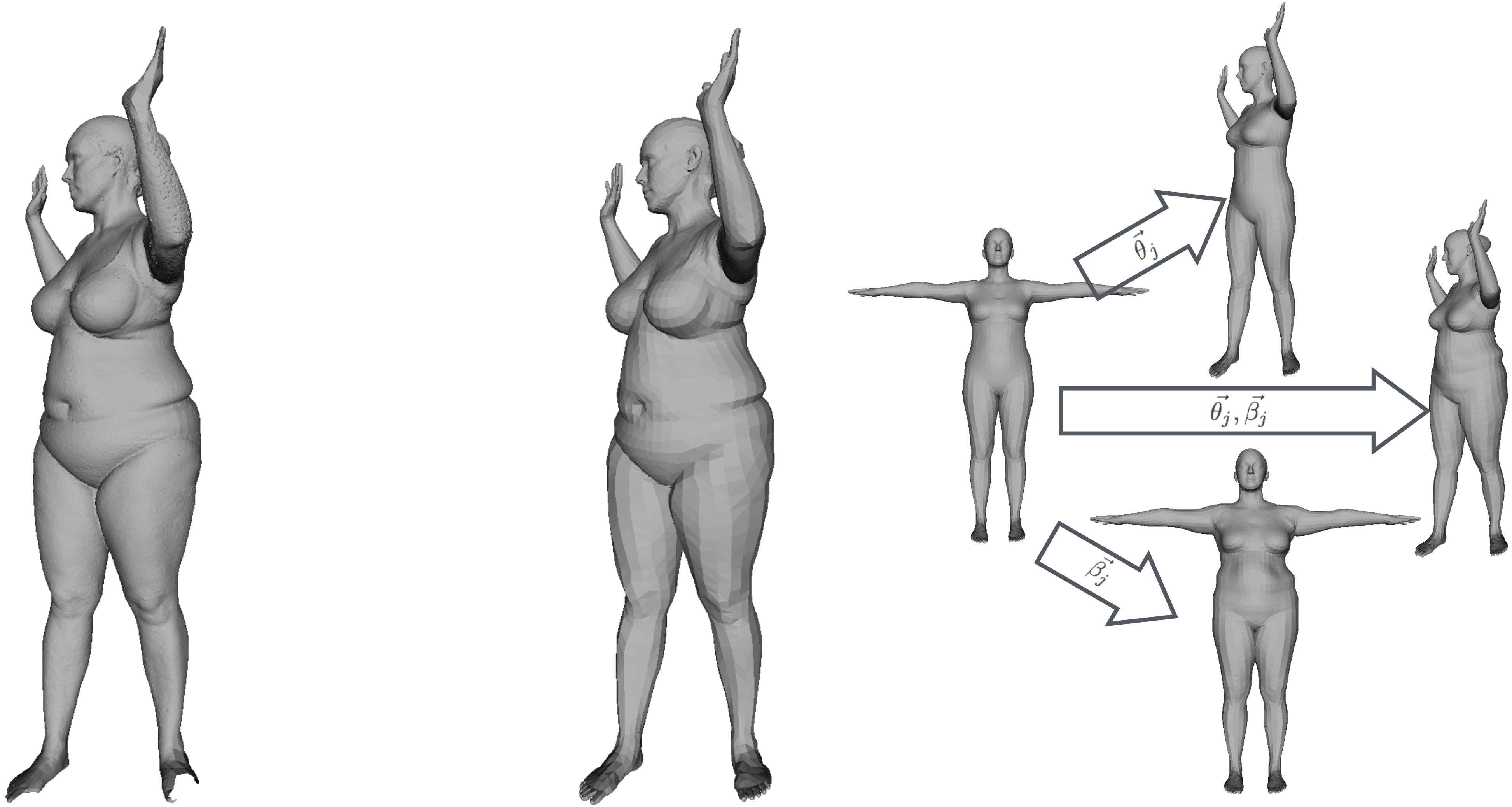
Training

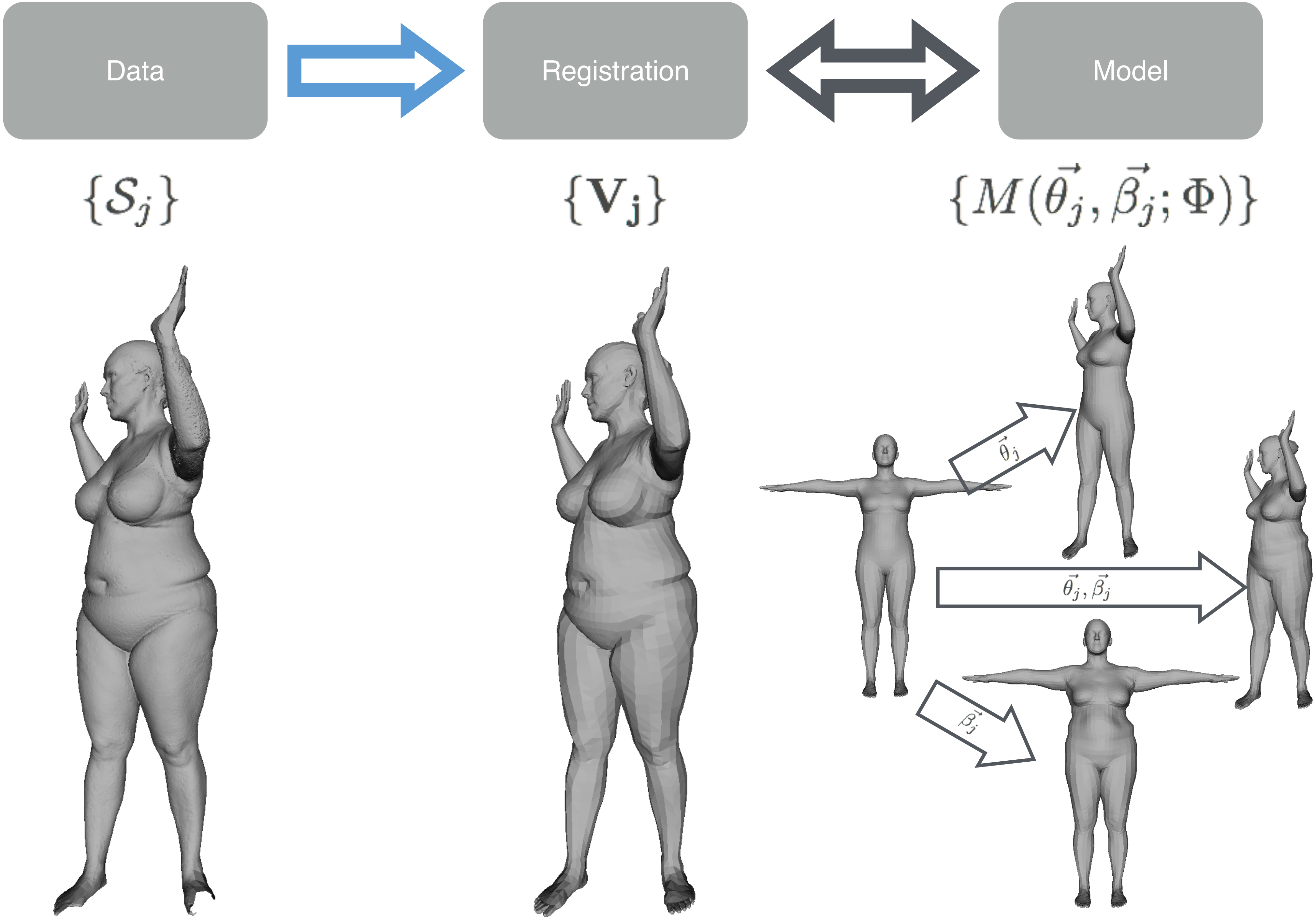
$$\Phi = \arg\min_{\Phi} \sum_j \min_{\vec{\theta}_j, \vec{\beta}_j} \| M(\vec{\theta}_j, \vec{\beta}_j; \Phi) - \mathbf{V}_j \|^2$$

Set of registrations

Data → Registration ↔ Model

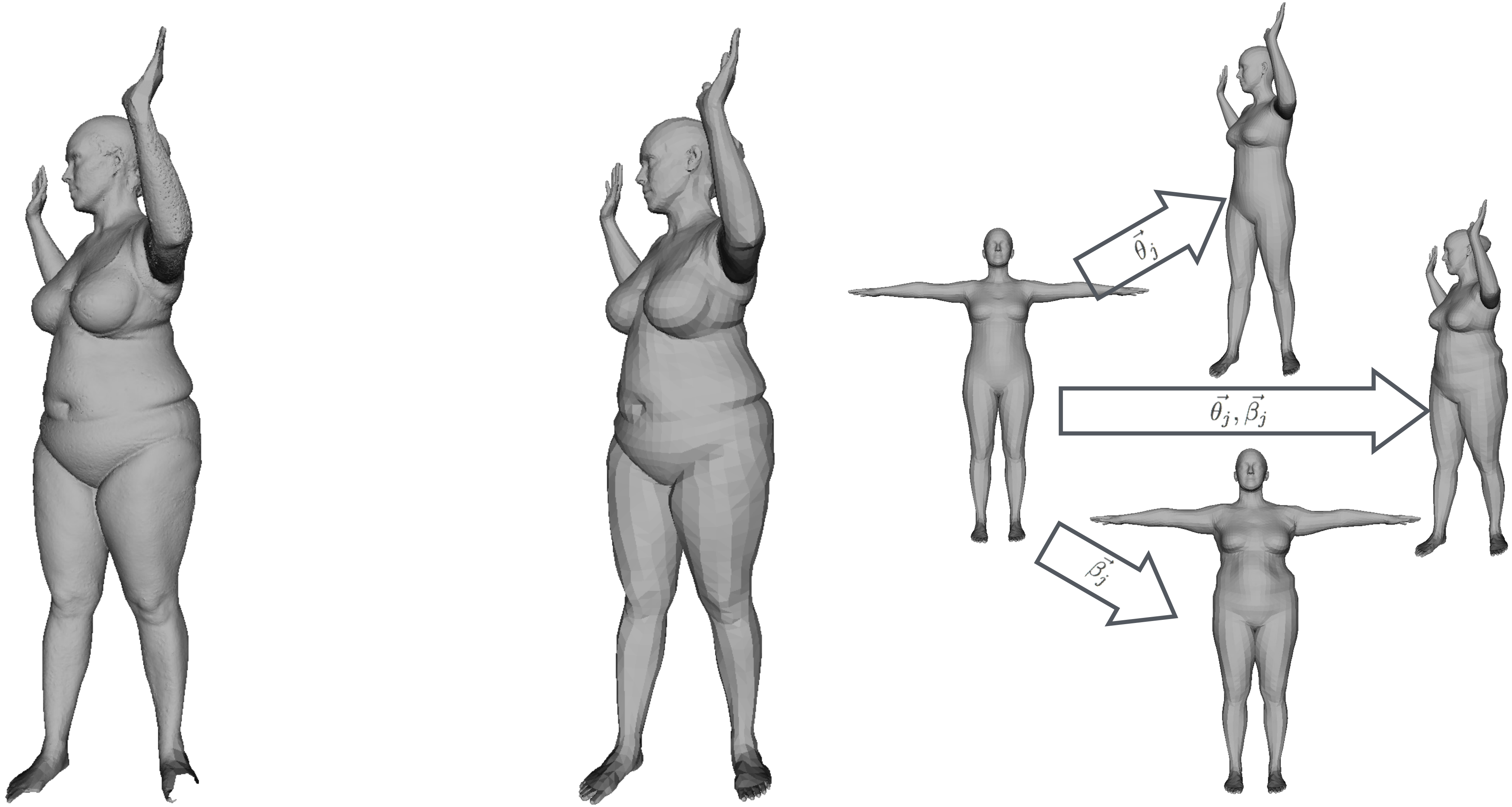
$\{\mathcal{S}_j\}$ $\{\mathbf{V_j}\}$ $\{M(\vec{\theta}_j, \vec{\beta}_j; \Phi)\}$

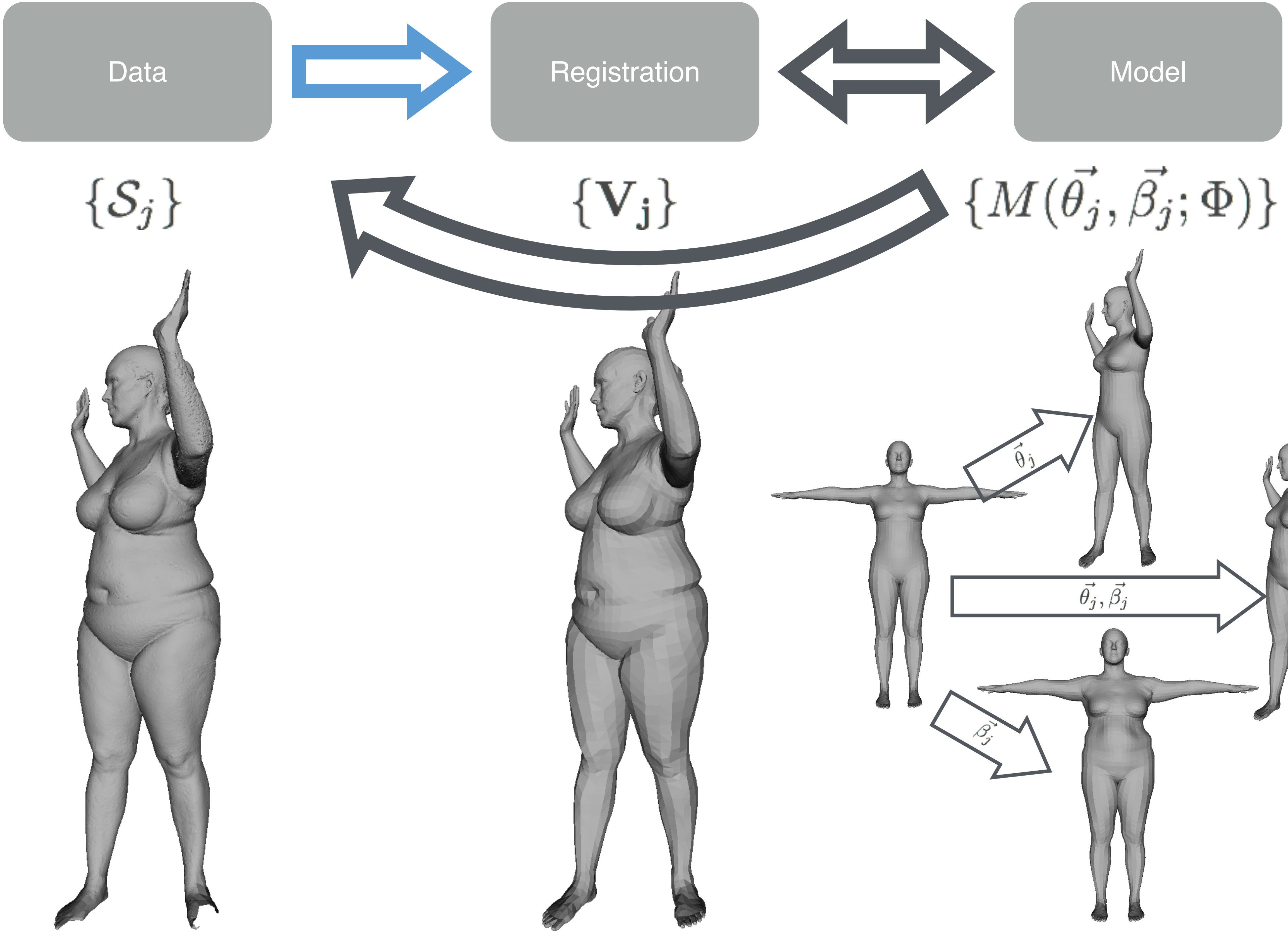



Data
Registration
Model
$\{S_j\}$
$\{\mathbf{V_j}\}$
$\{M(\vec{\theta}_j, \vec{\beta}_j; \Phi)\}$
$\vec{\theta}_j$
$\vec{\theta}_j, \vec{\beta}_j$
$\vec{\beta}_j$

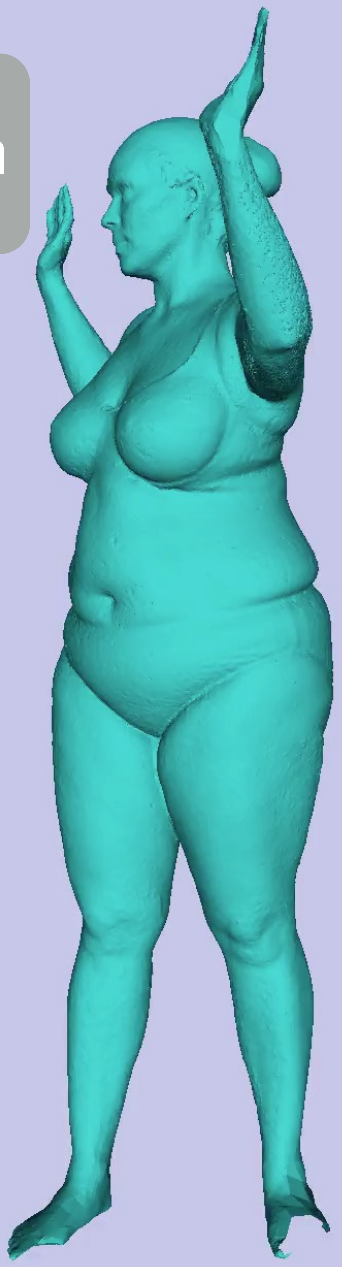
Data → Registration ↔ Model

$\{\mathcal{S}_j\}$ $\{\mathbf{V_j}\}$ $\{M(\vec{\theta}_j, \vec{\beta}_j; \Phi)\}$




Data
Registration
Model
$\{S_j\}$
$\{\mathbf{V_j}\}$
$\{M(\vec{\theta}_j, \vec{\beta}_j; \Phi)\}$
$\vec{\theta}_j$
$\vec{\theta}_j, \vec{\beta}_j$
$\vec{\beta}_j$

Data → pose + shape → Registration



$$\mathbf{V}_j = \arg\min_{\mathbf{V}_j}(d(\mathcal{S}_j, \mathcal{V}_j)$$

Refresher on ICP

1. Initialize

$$f^0 = \{\mathbf{R} = \mathbf{I}, \mathbf{t} = \frac{\sum \mathbf{y}_i}{N} - \frac{\sum \mathbf{x}_i}{N}, s = 1\}$$

2. Compute correspondences according to current best transform

$$\mathbf{x}_i^{j+1} = \arg\min_{\mathbf{x} \in \mathbf{X}} \|f^j(\mathbf{x}) - \mathbf{y}_i\|^2$$

3. Compute optimal transformation ($\mathrm{s}, \mathbf{R}, \mathbf{t}$)with Procrustes

$$f^{j+1} = \arg\min_f \sum_i \|f(\mathbf{x}_i^{j+1}) - \mathbf{y}_i\|^2$$

4. Terminate if converged (error below a threshold), otherwise iterate

Refresher on ICP

- $f(\cdot)$ need not just be a rigid transformation $(s, \mathbf{R}, \mathbf{t})$.

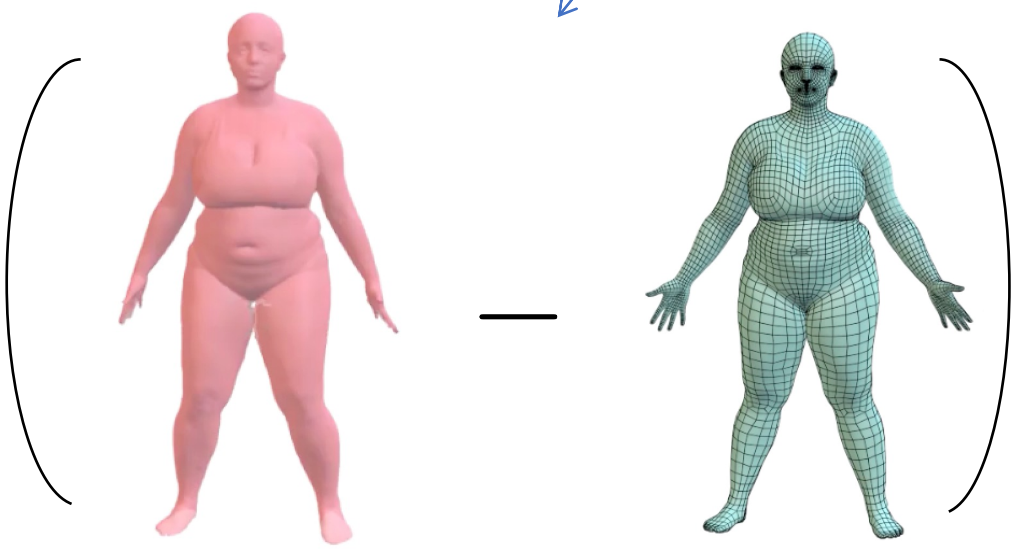
- $f(\cdot)$ can be any deformation model for the vertices V.

What should f() be?

- $f(\cdot)$ should it be all the degrees of freedom in **V**?

- $f(\cdot)$ should it be the SMPL function, parameterised by $(\vec{\theta}, \vec{\beta})$?

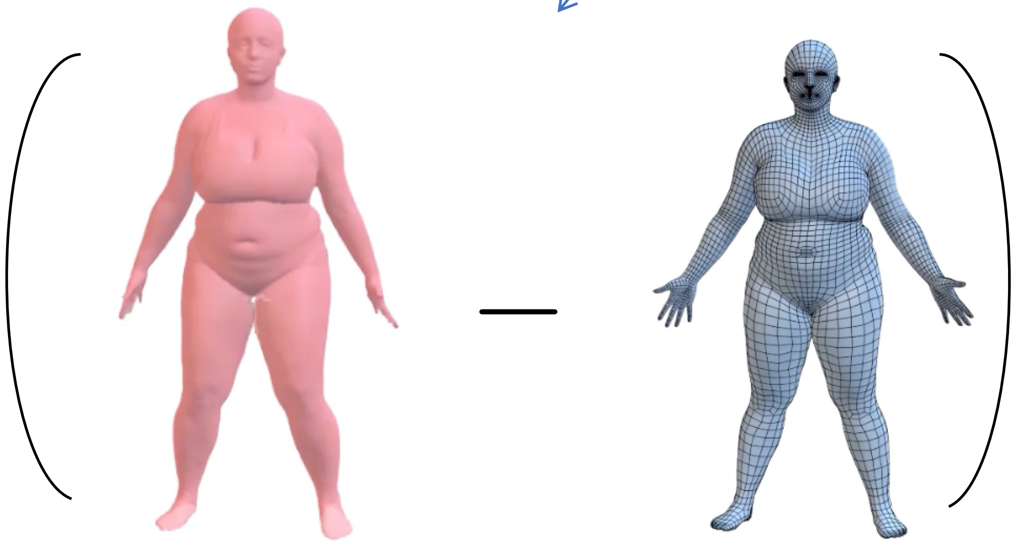
$$E(\mathbf{V}) = \sum_{\mathbf{s}_i \in \mathcal{S}} \mathrm{dist}(\mathbf{s}_i, \mathcal{V}(\mathbf{V})) + E_{\mathrm{prior}}(\mathbf{V})$$



Scan — Registration

- Optimizing **vertices directly is unstable**

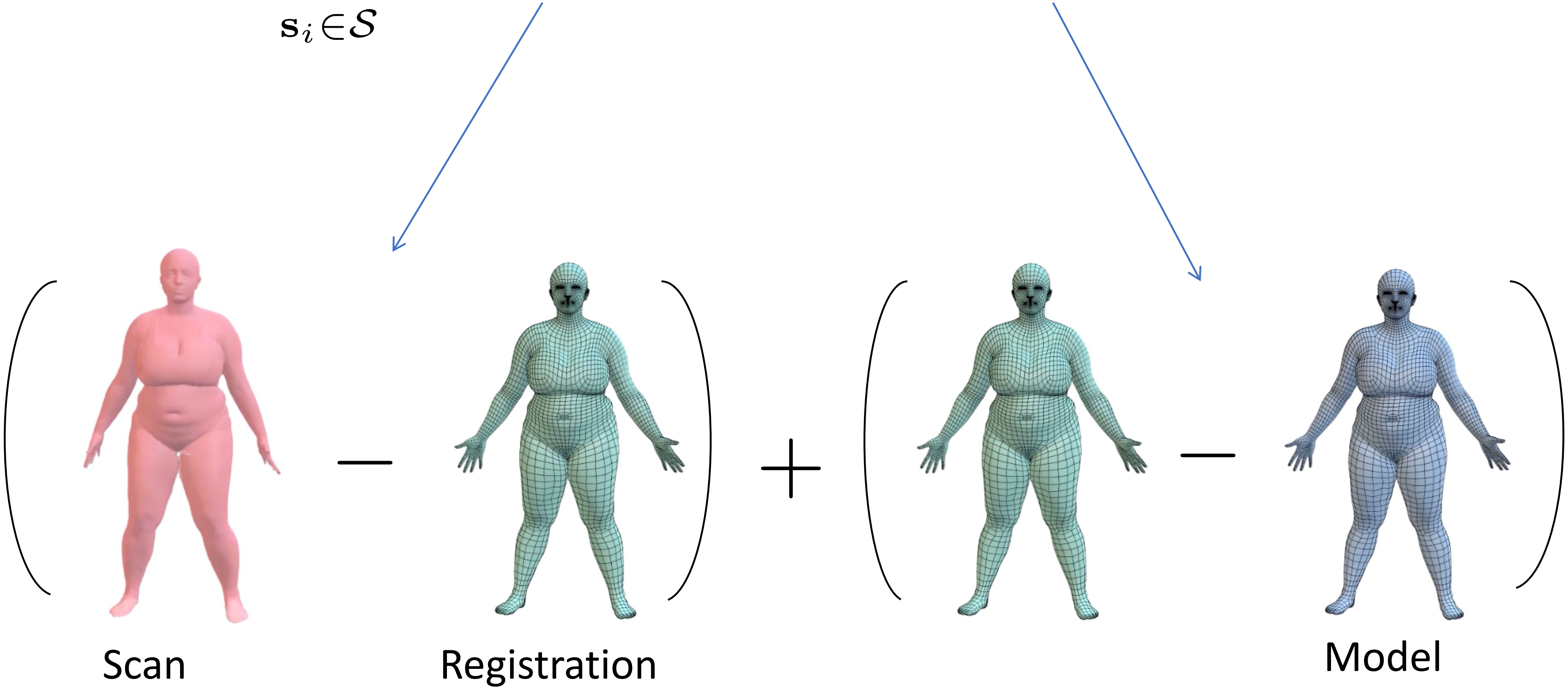
$$E(\theta, \beta) = \sum_{\mathbf{s}_i \in \mathcal{S}} \text{dist}(\mathbf{s}_i, \mathcal{M}(\theta, \beta)) + E_{\text{prior}}(\theta, \beta)$$



Scan Model

- Optimizing just the **model** does **not** capture detail.
- We **do not learn anything new** from the scan

$$E(\theta, \beta, \mathbf{V}) = \sum_{\mathbf{s}_i \in \mathcal{S}} \mathrm{dist}(\mathbf{s}_i, \mathcal{V}(\mathbf{V})) + \mathrm{dist}(\mathcal{V}(\mathbf{V}), \mathcal{M}(\theta, \beta)) + E_{\mathrm{prior}}(\theta, \beta)$$



Objective

$$\mathbf{V_j} = \arg\min_{\mathbf{V}_j}(\min_{\vec{\theta}_j,\vec{\beta}_j}(E_{reg}(\mathcal{S}_j, \mathbf{V}_j, \vec{\theta}_j, \vec{\beta}_j))$$

$$E_{reg}(\mathcal{S}_j, \mathbf{V}_j, \vec{\theta}_j, \vec{\beta}_j) =$$

Objective

$$\mathbf{V}_j = \arg\min_{\mathbf{V}_j}(\min_{\vec{\theta}_j,\vec{\beta}_j}(E_{reg}(\mathcal{S}_j, \mathbf{V}_j, \vec{\theta}_j, \vec{\beta}_j))$$

$$E_{reg}(\mathcal{S}_j, \mathbf{V}_j, \vec{\theta}_j, \vec{\beta}_j) = E_S(\mathcal{S}_j, \mathbf{V}_j) +$$

→ scan-to-mesh distance

Objective

$$\mathbf{V}_j = \arg\min_{\mathbf{V}_j}(\min_{\vec{\theta}_j,\vec{\beta}_j}(E_{reg}(\mathcal{S}_j, \mathbf{V}_j, \vec{\theta}_j, \vec{\beta}_j))$$

$$E_{reg}(\mathcal{S}_j, \mathbf{V}_j, \vec{\theta}_j, \vec{\beta}_j) = E_S(\mathcal{S}_j, \mathbf{V}_j) +$$

scan-to-mesh distance

$$\lambda_C E_C(\mathbf{V}_j, \vec{\theta}_j, \vec{\beta}_j) +$$

coupling

Objective

$$\mathbf{V}_j = \arg\min_{\mathbf{V}_j}\left(\min_{\vec{\theta}_j,\vec{\beta}_j}\left(E_{reg}(\mathcal{S}_j, \mathbf{V}_j, \vec{\theta}_j, \vec{\beta}_j)\right)\right)$$

$$E_{reg}(\mathcal{S}_j, \mathbf{V}_j, \vec{\theta}_j, \vec{\beta}_j) = E_S(\mathcal{S}_j, \mathbf{V}_j) +$$

$$\lambda_C E_C(\mathbf{V}_j, \vec{\theta}_j, \vec{\beta}_j) +$$

$$\lambda_\theta E_\theta(\vec{\theta}_j) +$$

scan-to-mesh distance

coupling

pose prior

Objective

$$\mathbf{V}_j = \arg\min_{\mathbf{V}_j}\left(\min_{\vec{\theta}_j, \vec{\beta}_j}\left(E_{reg}(\mathcal{S}_j, \mathbf{V}_j, \vec{\theta}_j, \vec{\beta}_j)\right)\right)$$

$$\begin{aligned} E_{reg}(\mathcal{S}_j, \mathbf{V}_j, \vec{\theta}_j, \vec{\beta}_j) = & E_S(\mathcal{S}_j, \mathbf{V}_j) + \\ & \lambda_C E_C(\mathbf{V}_j, \vec{\theta}_j, \vec{\beta}_j) + \\ & \lambda_\theta E_\theta(\vec{\theta}_j) + \\ & \lambda_\beta E_\beta(\vec{\beta}_j) \end{aligned}$$

- E_S: scan-to-mesh distance
- $\lambda_C E_C$: coupling
- $\lambda_\theta E_\theta$: pose prior
- $\lambda_\beta E_\beta$: shape prior

Objective

$$\mathbf{V}_j = \arg\min_{\mathbf{V}_j}(\min_{\vec{\theta}_j,\vec{\beta}_j}(E_{reg}(\mathcal{S}_j, \mathbf{V}_j, \vec{\theta}_j, \vec{\beta}_j))$$

$$E_{reg}(\mathcal{S}_j, \mathbf{V}_j, \vec{\theta}_j, \vec{\beta}_j) = E_S(\mathcal{S}_j, \mathbf{V}_j) +$$ → scan-to-mesh distance

$$\lambda_C E_C(\mathbf{V}_j, \vec{\theta}_j, \vec{\beta}_j) +$$ → coupling

$$\lambda_\theta E_\theta(\vec{\theta}_j) +$$ → pose prior

$$\lambda_\beta E_\beta(\vec{\beta}_j)$$ → shape prior

$\lambda_C, \lambda_\theta, \lambda_\beta$ → relative weights

Scan-to-mesh distance

$$E_S(\mathcal{S}_j, \mathbf{V}_j) = \sum_{\mathbf{s} \in \mathcal{S}_j} \rho \left(\min_{\mathbf{v} \in \mathcal{V}_j} \|\mathbf{s} - \mathbf{v}\| \right)$$

$$\rho(x) = \frac{x^2}{\sigma^2 + x^2}$$

Scan-to-mesh distance

$$E_S(\mathcal{S}_j, \mathbf{V}_j) = \sum_{\mathbf{s} \in \mathcal{S}_j} \rho \left(\min_{\mathbf{v} \in \mathcal{V}_j} \|\mathbf{s} - \mathbf{v}\| \right)$$

$$\rho(x) = \frac{x^2}{\sigma^2 + x^2}$$

Scan-to-mesh distance

$$E_S(\mathcal{S}_j, \mathbf{V}_j) = \sum_{\mathbf{s} \in \mathcal{S}_j} \rho\left(\min_{\mathbf{v} \in \mathcal{V}_j} \|\mathbf{s} - \mathbf{v}\|\right)$$

$$\rho(x) = \frac{x^2}{\sigma^2 + x^2}$$

Scan-to-mesh distance

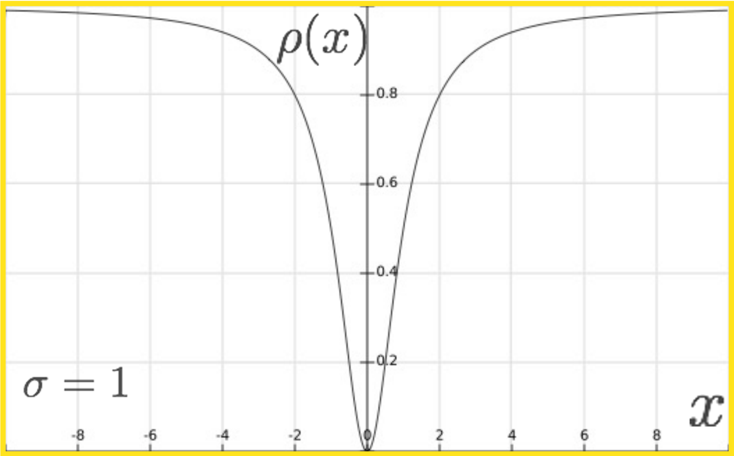
$$E_S(\mathcal{S}_j, \mathbf{V}_j) = \sum_{\mathbf{s} \in \mathcal{S}_j} \rho \left(\min_{\mathbf{v} \in \mathcal{V}_j} \|\mathbf{s} - \mathbf{v}\| \right)$$

$$\rho(x) = \frac{x^2}{\sigma^2 + x^2}$$

Scan-to-mesh distance

$$E_S(\mathcal{S}_j, \mathbf{V}_j) = \sum_{\mathbf{s} \in \mathcal{S}_j} \rho\left(\min_{\mathbf{v} \in \mathcal{V}_j} \|\mathbf{s} - \mathbf{v}\| \right)$$

$$\rho(x) = \frac{x^2}{\sigma^2 + x^2}$$



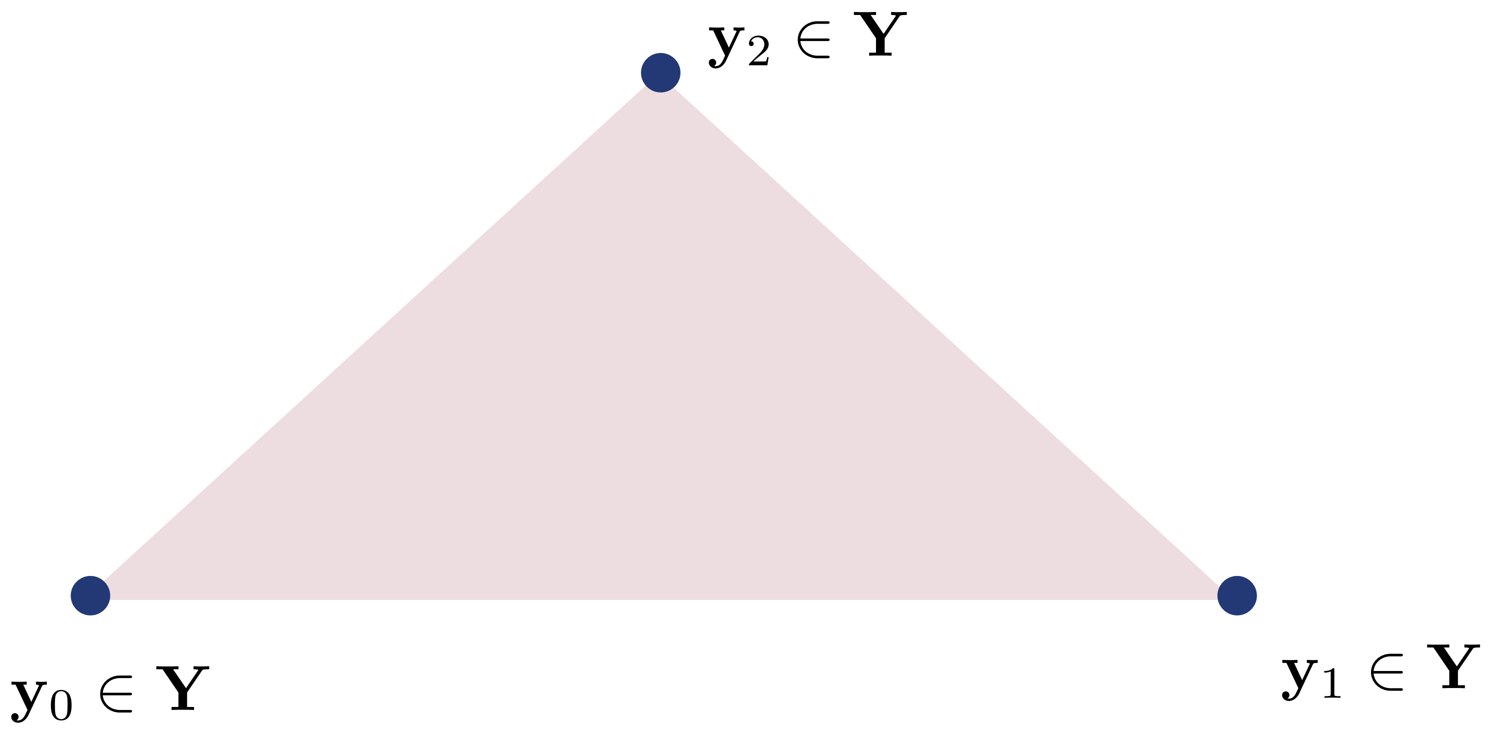
Scan-to-mesh distance

$$E_S(\mathcal{S}_j, \mathbf{V}_j) = \sum_{\mathbf{s} \in \mathcal{S}_j} \rho\left(\min_{\mathbf{v} \in \mathcal{V}_j} \|\mathbf{s} - \mathbf{v}\| \right)$$

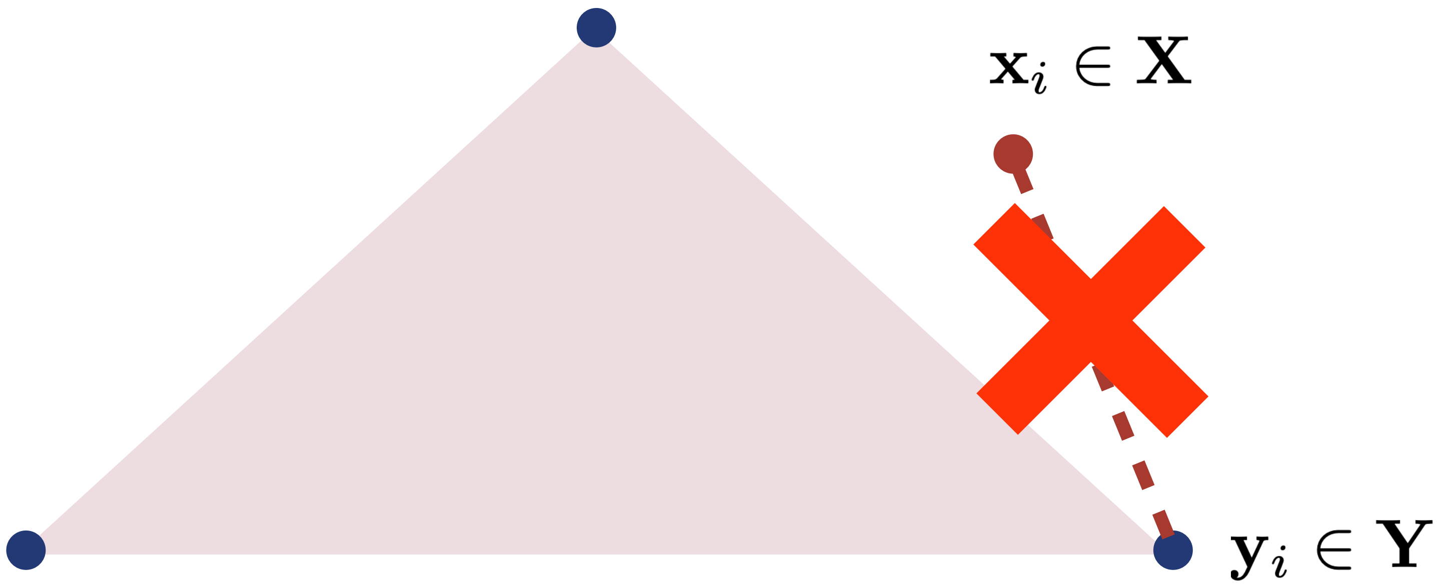
$$\rho(x) = \frac{x^2}{\sigma^2 + x^2}$$

point-to-surface distance!

A much better objective: Point-to-surface distance

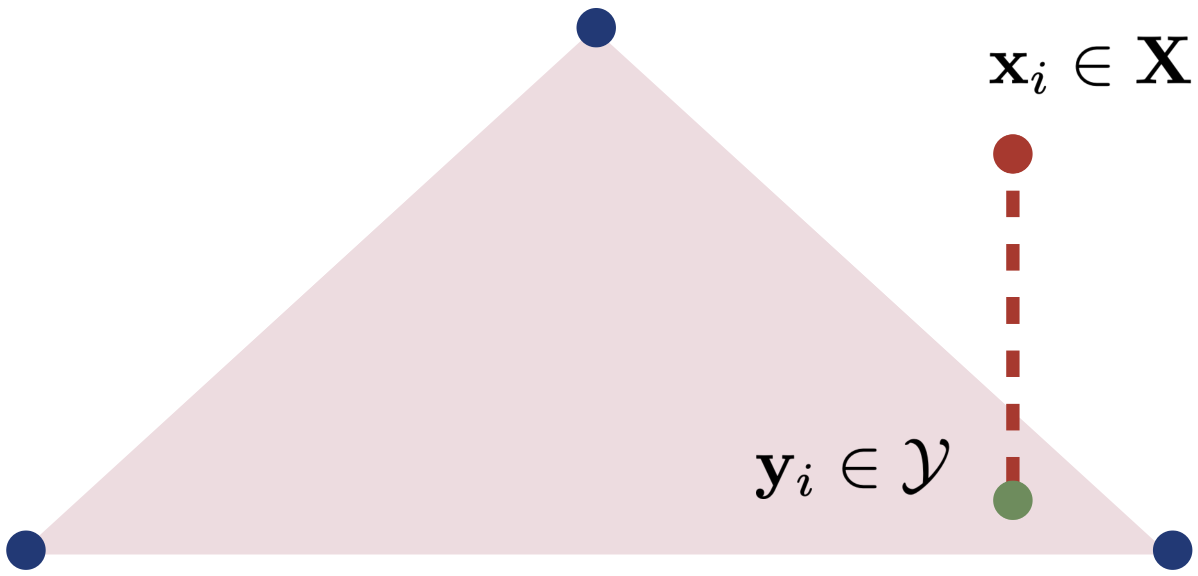


Closest points: avoid local minima



Point-to-point distance

Closest points: avoid local minima



Point-to-surface distance

Coupling

$$E_C(\mathbf{V}_j, \vec{\theta}_j, \vec{\beta}_j) \equiv \mathrm{dist}(\mathbf{V}_j, M(\vec{\theta}_j, \vec{\beta}_j))$$

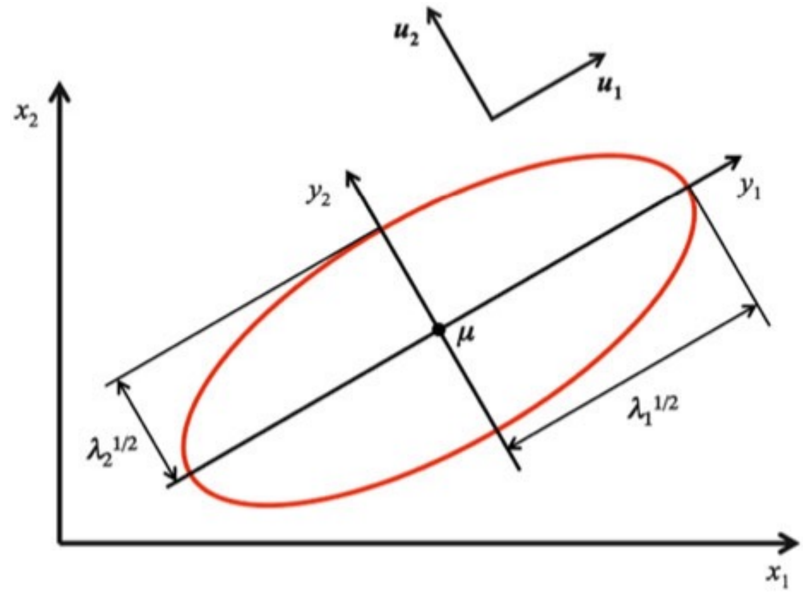
Coupling

$$E_C(\mathbf{V}_j, \vec{\theta_j}, \vec{\beta_j}) = \begin{cases} \|\mathbf{V}_j - M(\vec{\theta_j}, \vec{\beta_j})\|_F^2, & \text{if coupling on vertices} \\ \end{cases}$$

Coupling

$$E_C(\mathbf{V}_j, \vec{\theta_j}, \vec{\beta_j}) = \begin{cases} \|\mathbf{V}_j - M(\vec{\theta_j}, \vec{\beta_j})\|_F^2, & \text{if coupling on vertices} \\ \|A\mathbf{V}_j - AM(\vec{\theta_j}, \vec{\beta_j})\|_F^2, & \text{if coupling on edges} \end{cases}$$

Priors

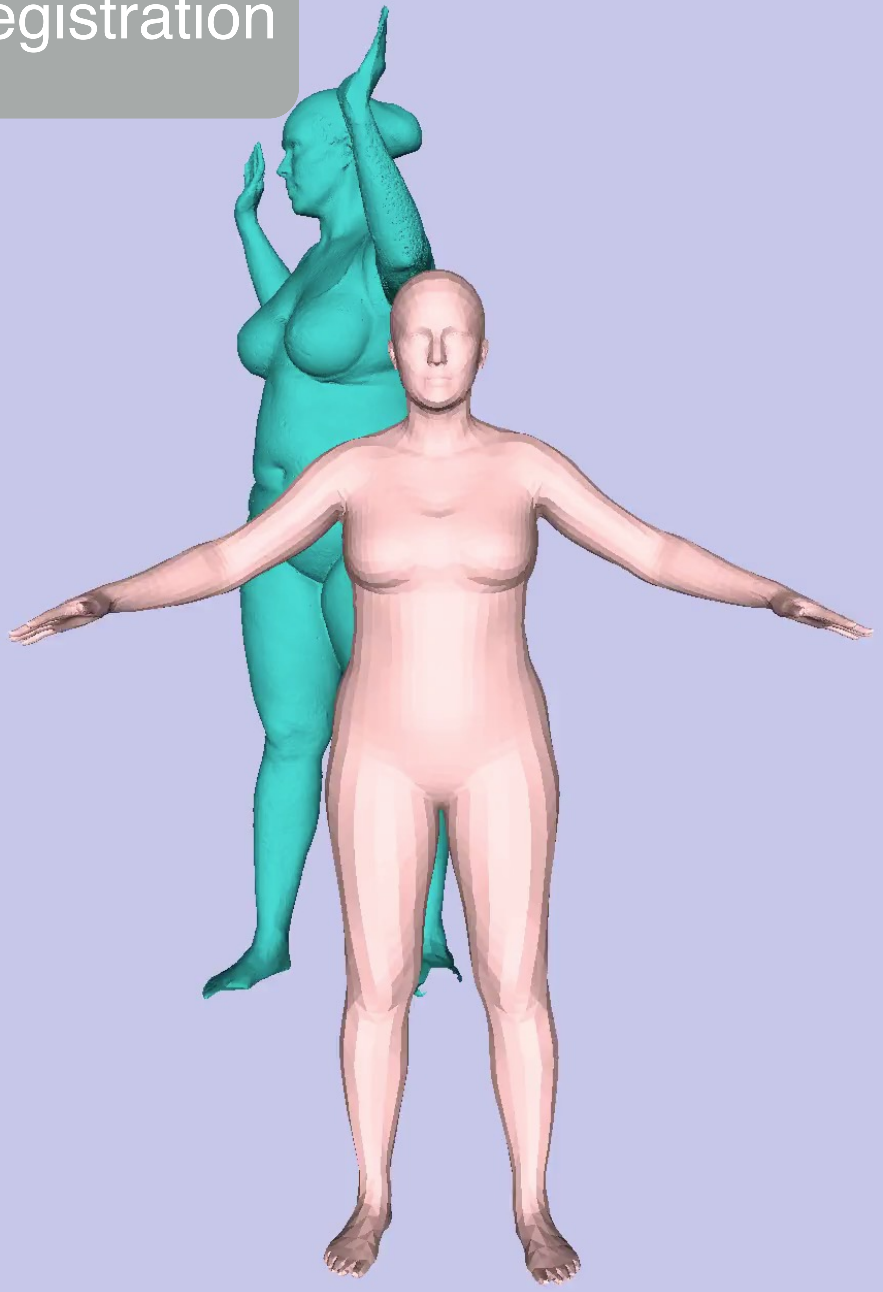


- Priors are based on Mahalanobis distance
- Assumes a Gaussian Distribution

$$E_\theta(\vec{\theta}_j) = (\vec{\theta}_j - \vec{\mu}_\theta)\Sigma_\theta^{-1}(\vec{\theta}_j - \vec{\mu}_\theta)^T$$

$$E_\beta(\vec{\beta}_j) = (\vec{\beta}_j - \vec{\mu}_\beta)\Sigma_\beta^{-1}(\vec{\beta}_j - \vec{\mu}_\beta)^T$$

Data → pose + shape → Registration



Stage 1

$$\lambda_\theta >> 1$$
$$\lambda_\beta >> 1$$
$$\lambda_C = \inf$$

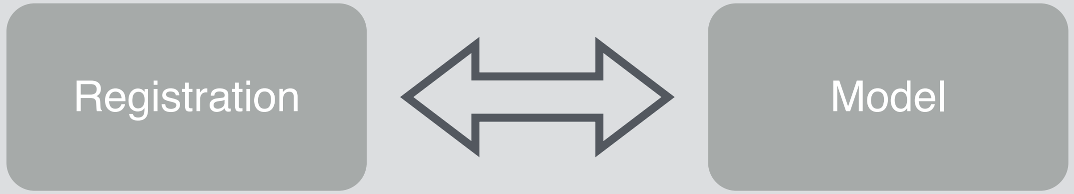
Stage 2

$$\lambda_\theta \approx 1$$
$$\lambda_\beta \approx 1$$
$$\lambda_C >> 1$$

Stage 3

$$\lambda_\theta \approx 0$$
$$\lambda_\beta \approx 0$$
$$\lambda_C \approx 1$$

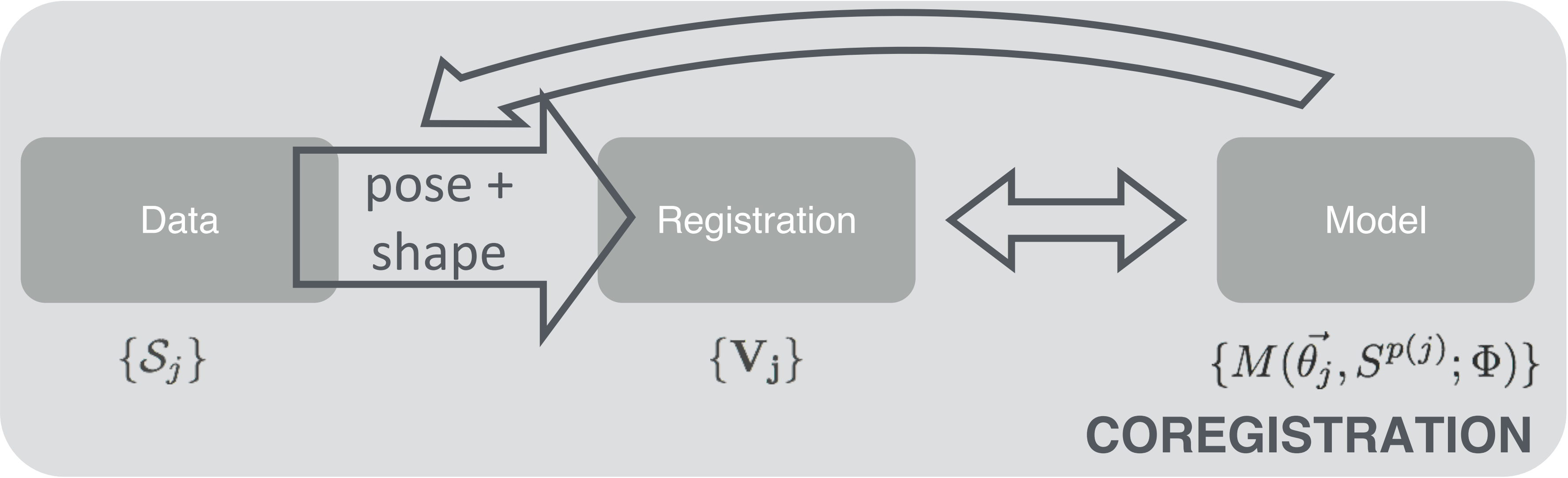
Given Registrations I can train a Model



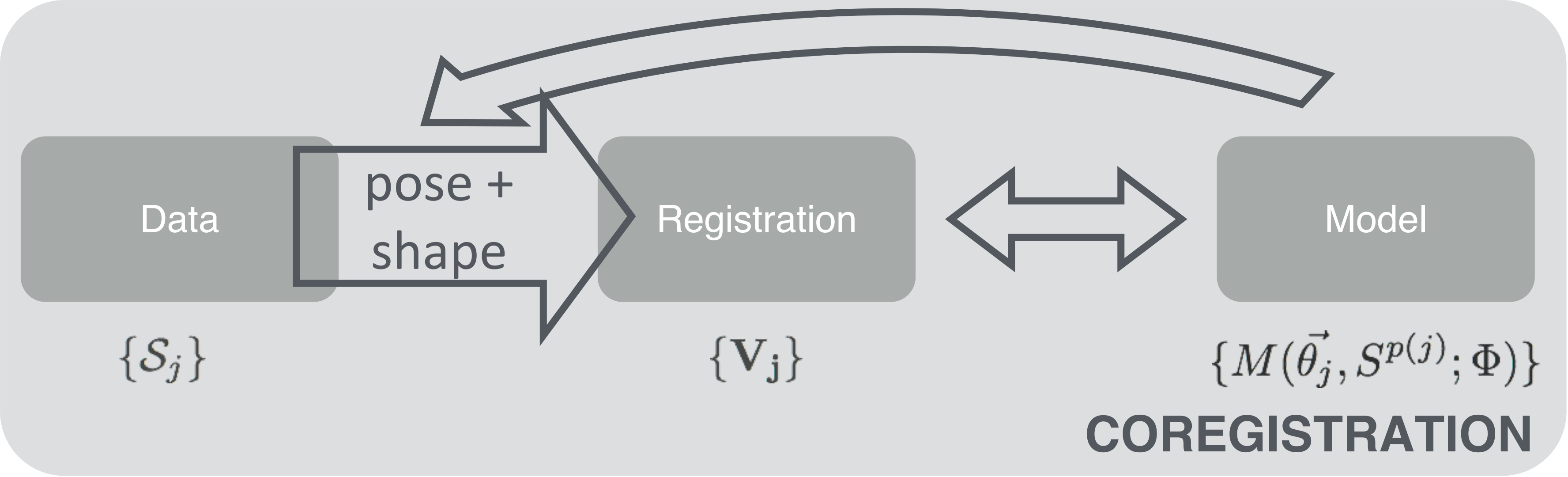
$\{\mathbf{V_j}\}$

$\{M(\vec{\theta}_j, \vec{\beta}_j; \Phi)\}$

Given a Model I can obtain registrations



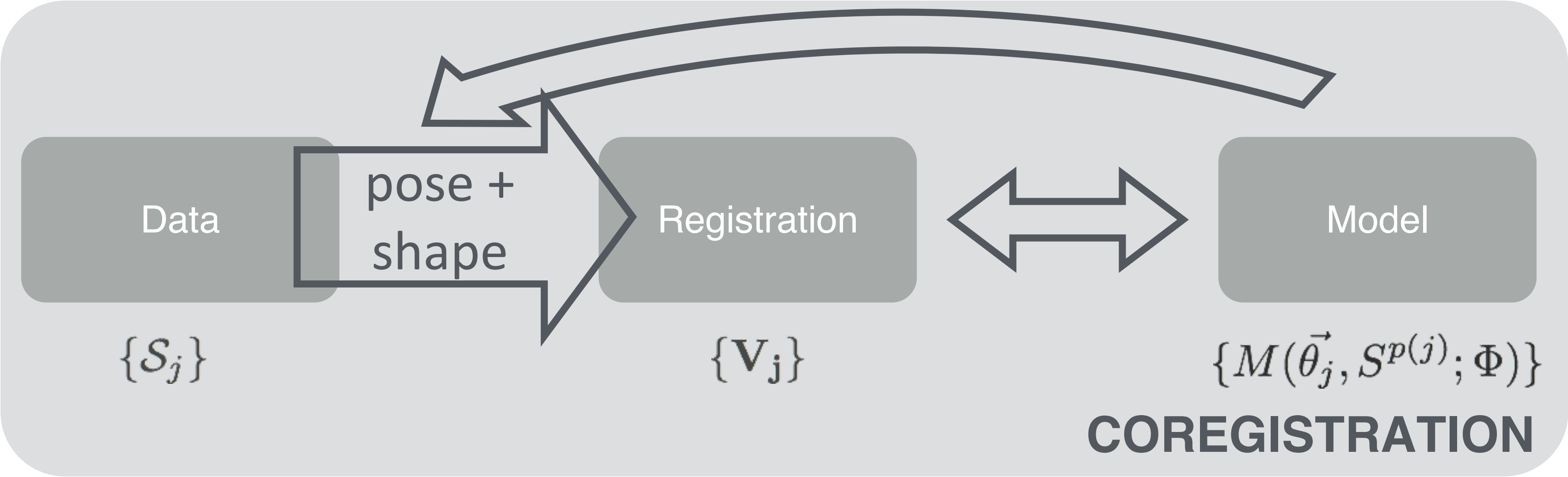
But, how can I solve both problems at once?



Joint registration and model learning

The key idea is to minimize the registration objective (loss function)

$$
\begin{aligned}
E_{reg}(\mathcal{S}_j, \mathbf{V}_j, \vec{\theta}_j, \vec{\beta}_j) = & E_S(\mathcal{S}_j, \mathbf{V}_j) + \\
& \lambda_C E_C(\mathbf{V}_j, \vec{\theta}_j, \vec{\beta}_j) + \\
& \lambda_\theta E_\theta(\vec{\theta}_j) + \\
& \lambda_\beta E_\beta(\vec{\beta}_j)
\end{aligned}
$$



The key idea is to **minimize** the registration objective across **the dataset of scans**

$$E(\{\mathcal{S}_j\}, \{\mathbf{V}_j\}, \{\vec{\theta_j}\}, \Phi) = \sum_j (E_S(\mathcal{S}_j, \mathbf{V}_j) +$$

$$\lambda_C E_C(\mathbf{V}_j, \vec{\theta_j}, \Phi) +$$

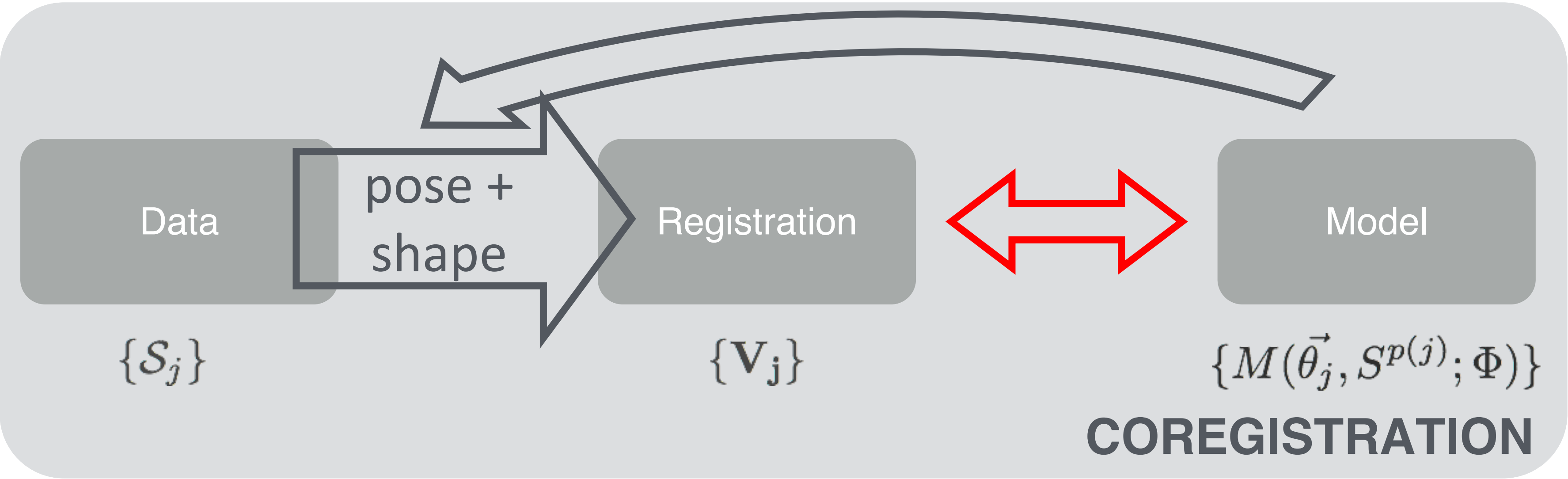
$$\lambda_\theta E_\theta(\vec{\theta_j})) +$$

$$\lambda_\Phi E_\Phi(\Phi)$$

Sums over dataset scans!

model regularisation

Hirshberg et al., Coregistration: Simultaneous alignment and modeling of articulated 3D shape, ECCV'12



$$\Phi = \arg\min_{\Phi}(E(\{\mathcal{S}_j\}, \{\mathbf{V}_j\}, \{\vec{\theta}_j, \vec{\beta}\}, \Phi)) \text{ with } \{\mathbf{V}_j\} \text{ fixed}$$

$$\{\mathbf{V}_j\} = \arg\min_{\{\mathbf{V}_j\}}(E(\{\mathcal{S}_j\}, \{\mathbf{V}_j\}, \{\vec{\theta}_j, \vec{\beta}_j\}, \Phi)) \text{ with } \Phi \text{ fixed}$$

Hirshberg et al., Coregistration: Simultaneous alignment and modeling of articulated 3D shape, ECCV'12

Summary

- SCAPE based models are based on triangle deformations and stitching triangles
- Training a model requires solving for model hyper-parameters and shape and pose for each registration
- Obtaining registrations requires a model. Obtaining a model requires registrations.
- This chicken and egg problem can be addressed with co-registration (similar to EM)

Summary

- SCAPE based models are based on triangle deformations and stitching triangles
- Training a model requires solving for model hyper-parameters and shape and pose for each registration
- Obtaining registrations requires a model. Obtaining a model requires registrations.
- This chicken and egg problem can be addressed with co-registration (similar to EM)

Summary

- SCAPE based models are based on triangle deformations and stitching triangles
- Training a model requires solving for model hyper-parameters and shape and pose for each registration
- Obtaining registrations requires a model. Obtaining a model requires registrations.
- This chicken and egg problem can be addressed with co-registration (similar to EM)

Summary

- SCAPE based models are based on triangle deformations and stitching triangles
- Training a model requires solving for model hyper-parameters and shape and pose for each registration
- Obtaining registrations requires a model. Obtaining a model requires registrations.
- This chicken and egg problem can be addressed with co-registration (similar to EM)

Slide Credits and resources

- Slides adapted from Learning Human Shapes in Motion. Siggraph tutorial 2016. Special thanks to Javier Romero