# Virtual Humans – Winter 23/24

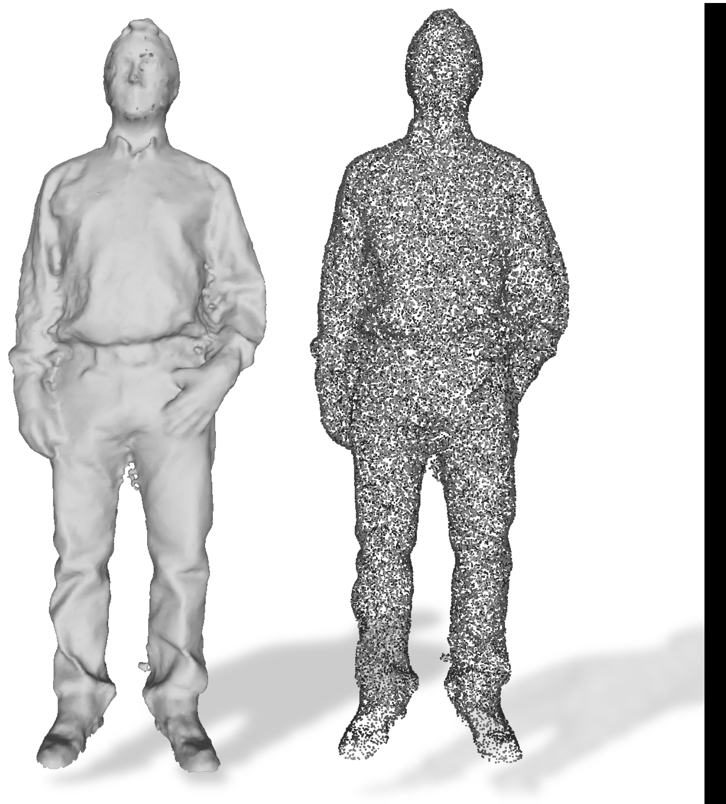
Lecture 5_2 – Learning based registration

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

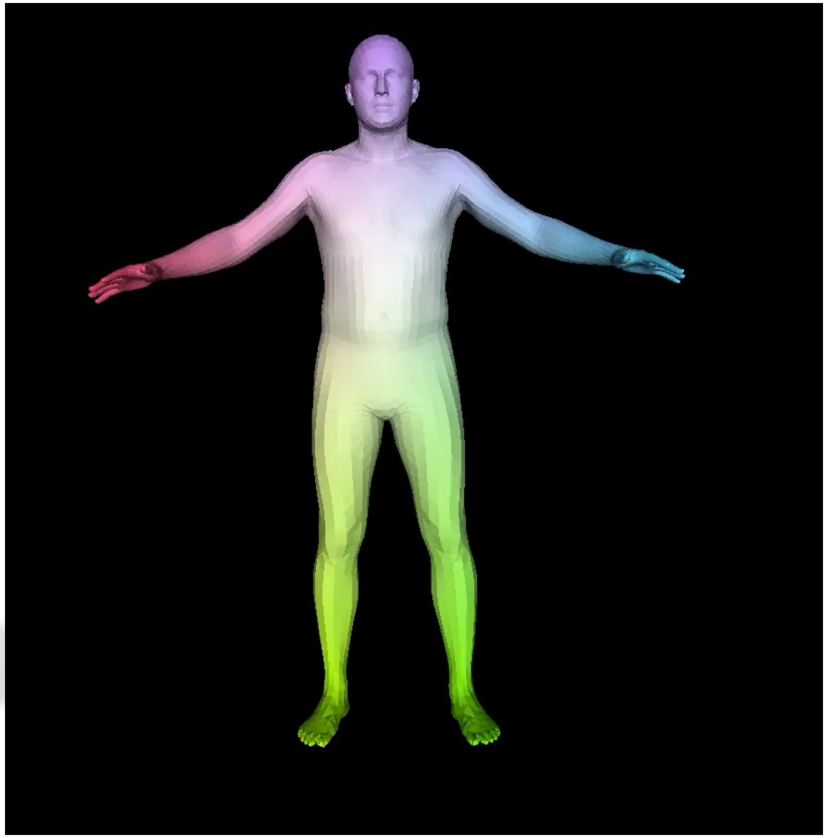
EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

# 3D scan → Human Model

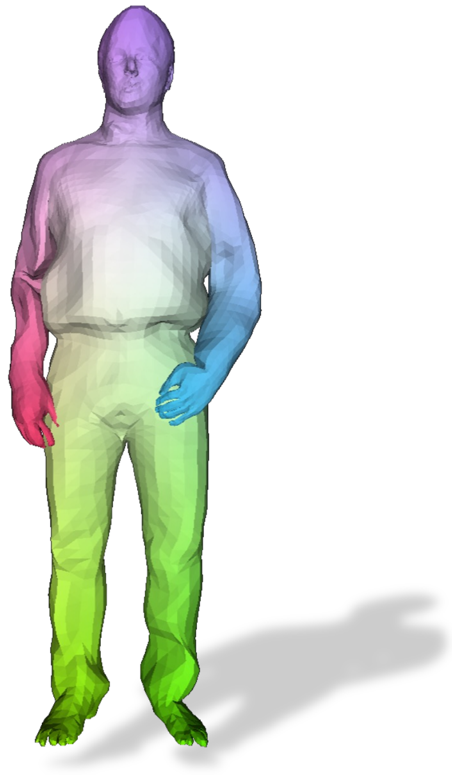
Input: 3D scan/ Pointcloud



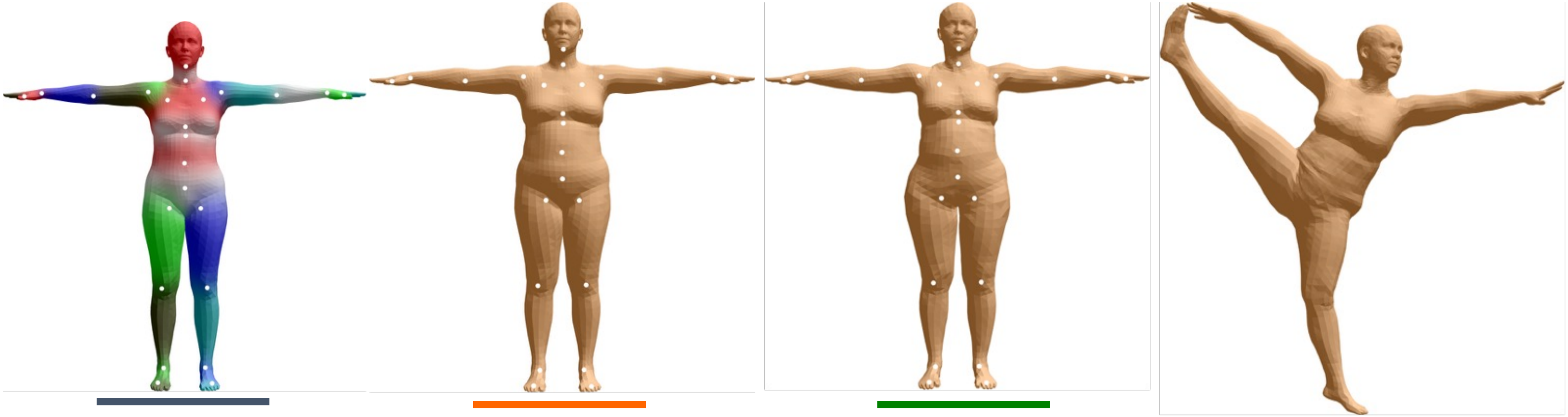
Colour coded SMPL model



Output: Registered SMPL+D



# SMPL model



$$T(\boldsymbol{\theta}, \boldsymbol{\beta}) = \underline{\mathbf{T}_\mu} + \underline{B_s(\boldsymbol{\beta})} + \underline{B_p(\boldsymbol{\theta})}$$

Vertices in a 0-pose

# SMPL + Clothing

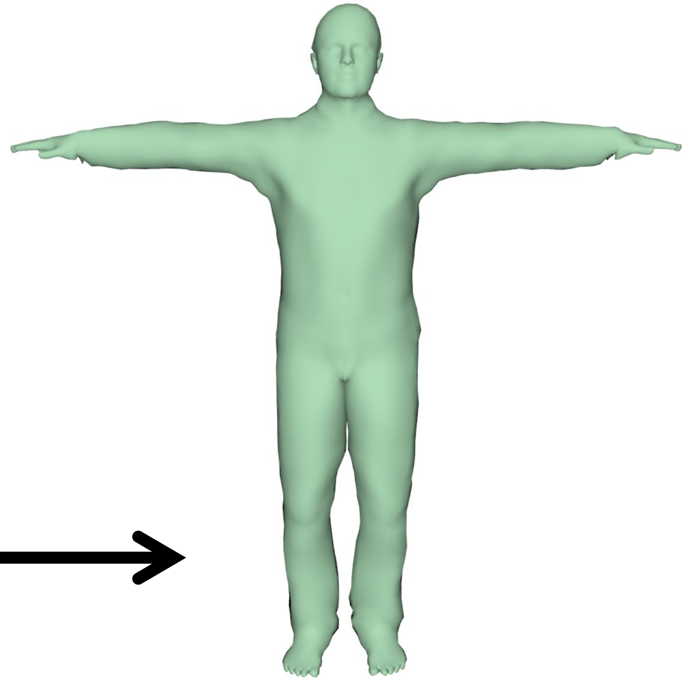
Vertices in a 0-pose

$$T(\boldsymbol{\theta}, \boldsymbol{\beta}, \mathbf{D}) = \mathbf{T}_{\mu} + B_{\boldsymbol{s}}(\boldsymbol{\beta}) + B_{\boldsymbol{p}}(\boldsymbol{\theta}) + \mathbf{D}$$

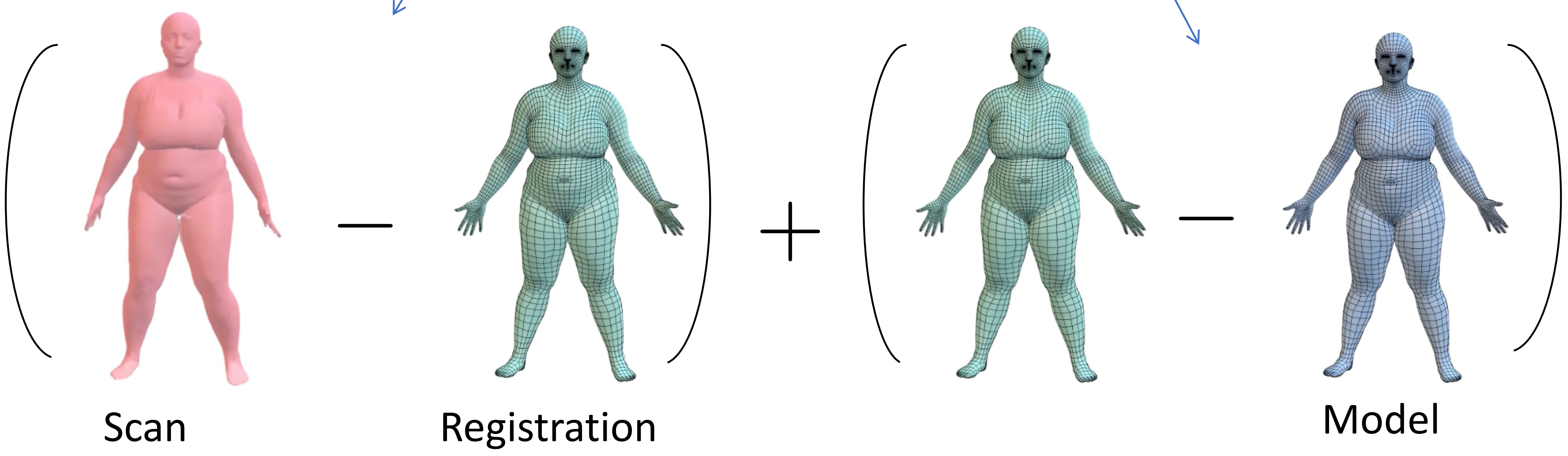
$\boldsymbol{\theta}$ Pose parameters

$\boldsymbol{\beta}$ Shape parameters

$\mathbf{D}$ Personal details + clothing



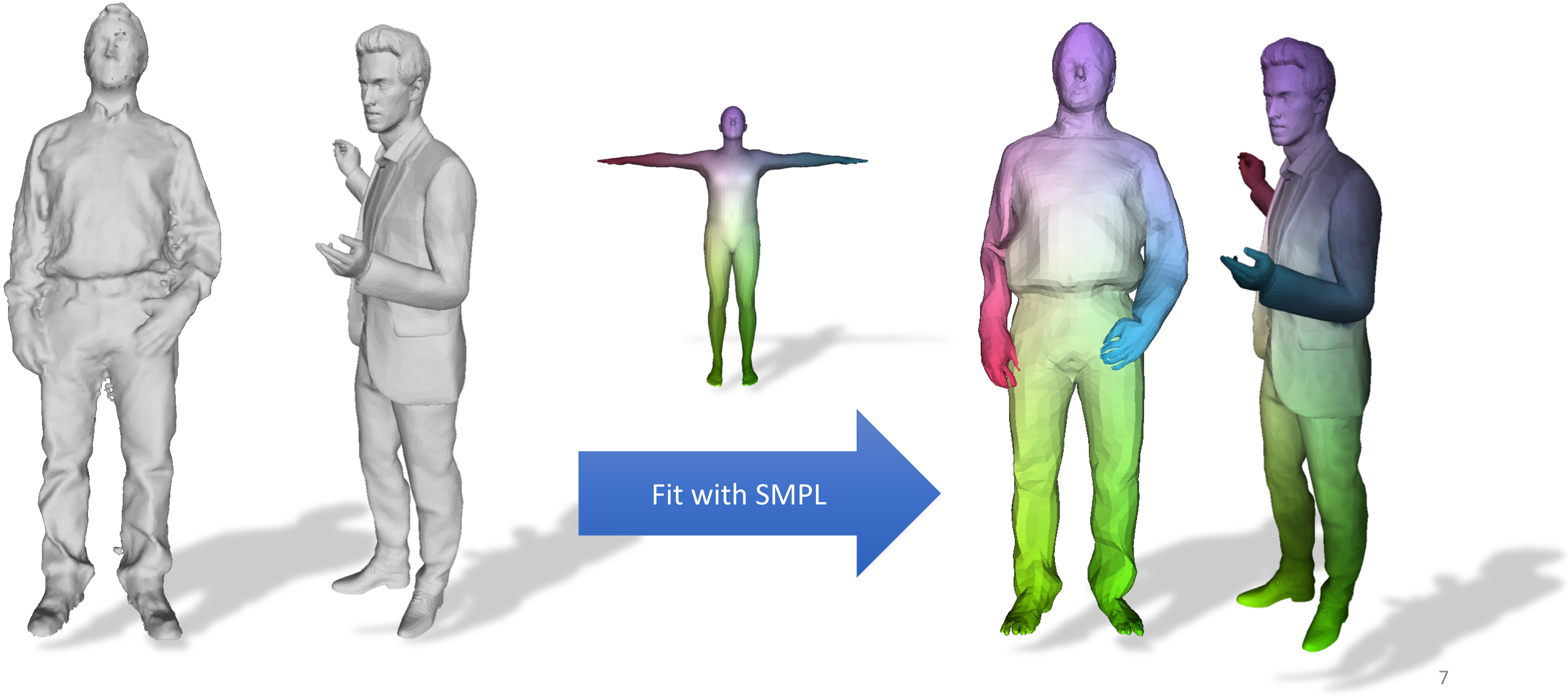
$$E(\theta, \beta, \mathbf{V}) = \sum_{\mathbf{s}_i \in \mathcal{S}} \mathrm{dist}(\mathbf{s}_i, \mathcal{V}(\mathbf{V})) + \mathrm{dist}(\mathcal{V}(\mathbf{V}), \mathcal{M}(\theta, \beta)) + E_{\text{prior}}(\theta, \beta)$$



# Why fit SMPL to scans?

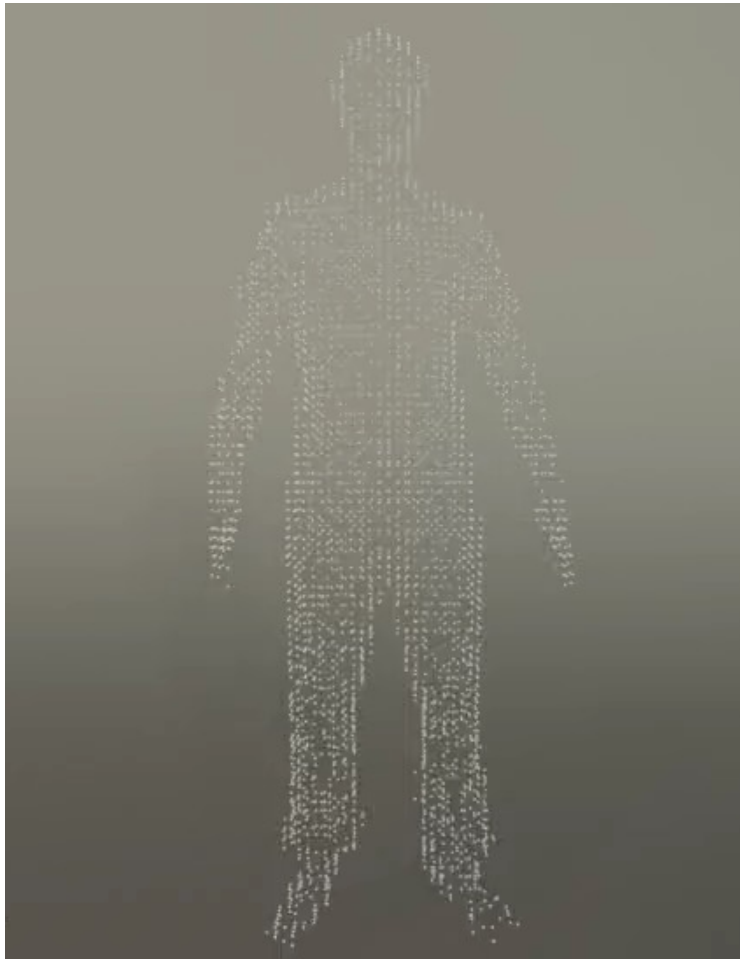
We motivated finding registration as a key ingredient to train a body model

# Find correspondences between meshes



# Tracking scans/ point clouds

Input PC seq.



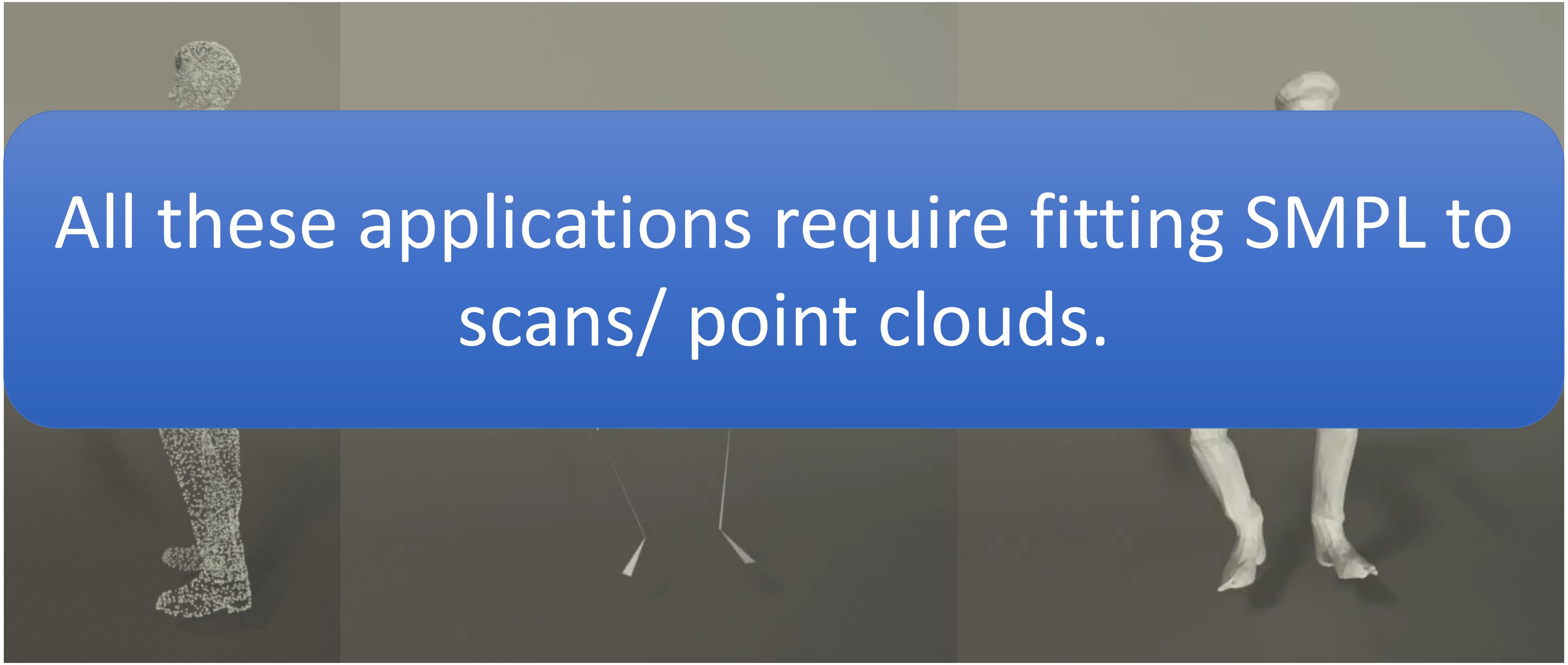
Tracked SMPL model

# Controlling static shapes

Input PC | Input pose sequence | Animated SMPL+D

# Controlling static shapes

Input PC | Input pose sequence | Animated SMPL+D



# Fit SMPL or SMPL+D to scans using ICP (compute registrations)

# Objective

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

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

- $E_S(\mathcal{S}_j, \mathbf{V}_j)$ → 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

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

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

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

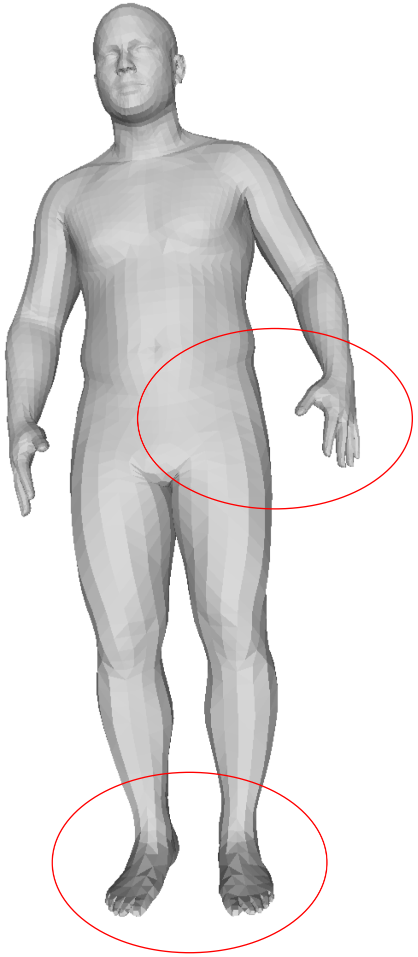
3. Compute optimal transformation ( $\mathrm{s}, \mathbf{R}, \mathbf{t}$ )with Procrustes $$f^{j+1} = \arg\min_f \sum_i \|f(\mathbf{x}_i^{j+1}) - \mathbf{y}_i\|^2$$

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

# Limitations of ICP

# Limitations of ICP

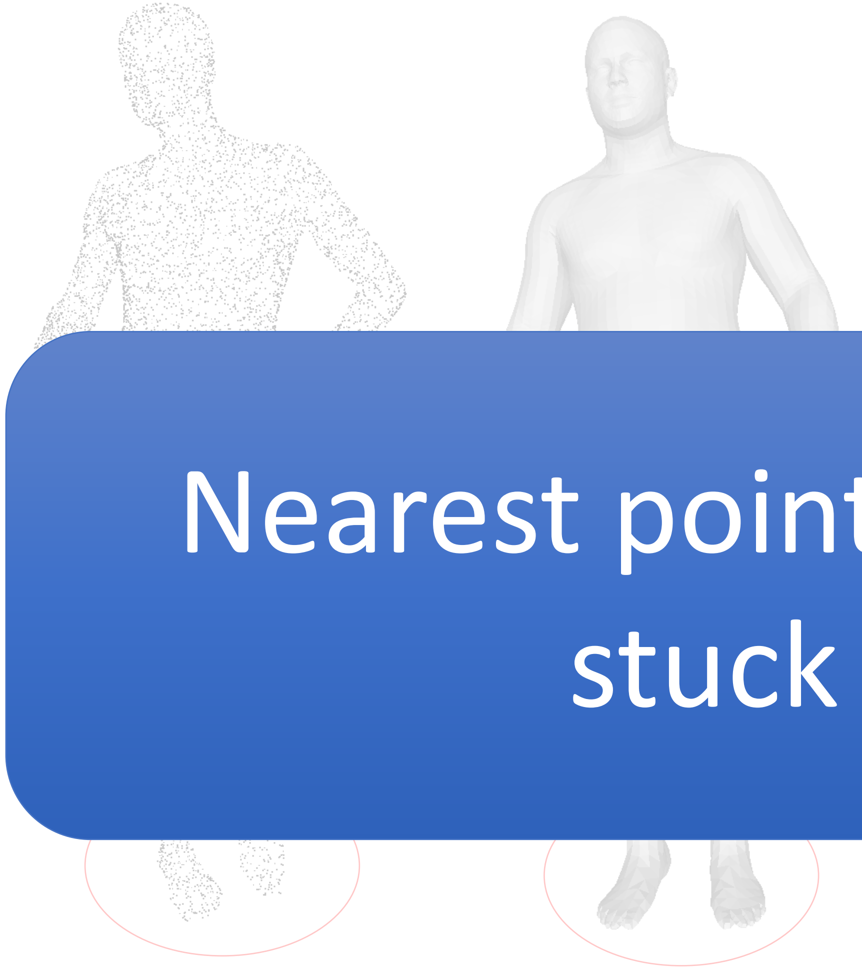
Input PC



SMPL fit - ICP

- ICP -> closest points can be wrong
- Doesn't distinguish if the correspondence is semantically correct.
- For example, pointcloud hand points are explained by the waist of the model

# Limitations of ICP



Input PC SMPL fit - ICP

– ICP cares about closest point.

– Doesn't distinguish if the

ed

**Nearest point as correspondence gets stuck in local minimas!**

# Learning based fitting

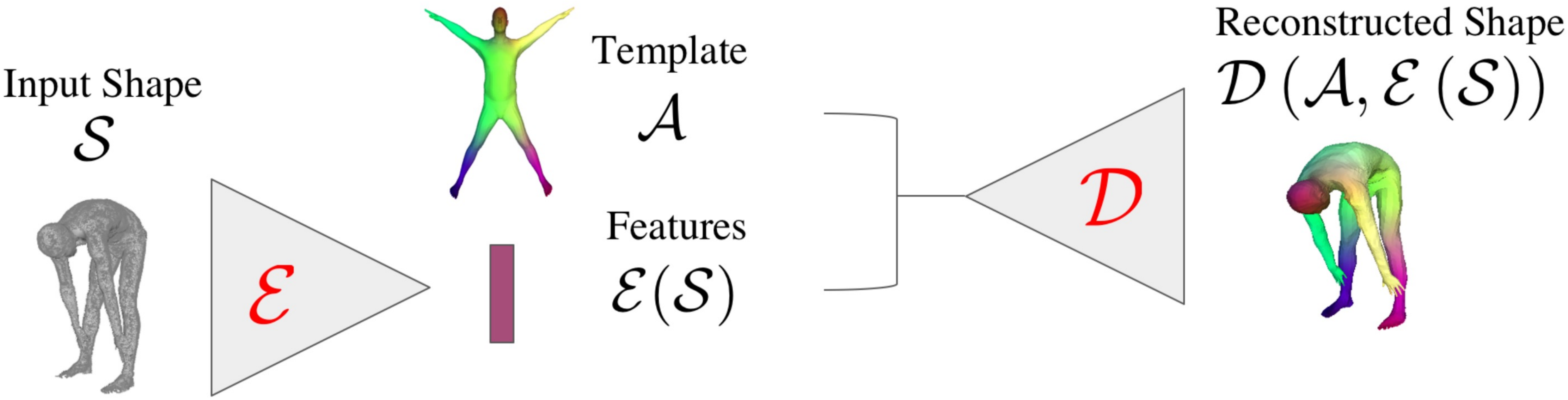
Can we use data to learn how to fit a template mesh to scan/ point cloud?

# Learning based fitting

- Non-parametric: Fit a template mesh to data.
  - 3D-CODED, Groueix et al. ECCV‘18
- Parametric: Fit a model to data.
  - IPNet, Bhatnagar et al. ECCV'20
  - LoopReg, Bhatnagar et al. NeurIPS’20
- Hybrid:
  - Learned Vertex Descent, Corona et al. ECCV’22
  - (we will see later in the course)

# Learn to deform vertices of a template

- Encode input shape into a feature vector.
- Directly predict locations of vertices of template.




3D-CODED, Groueix et al. ECCV'18

# Advantages/ Disadvantages

✓Learning based model, generalises better than ICP.

– Gets stuck in local minima. Need to init. ~100 global rots.

– No details.

– Registered template is not controllable!
Can't pose and shape.

**Bring back the parametric model!**

- Can we "learn" to fit SMPL model to data?
  - **Make scans controllable**

- Can we capture high frequency details?
  - **More realistic**

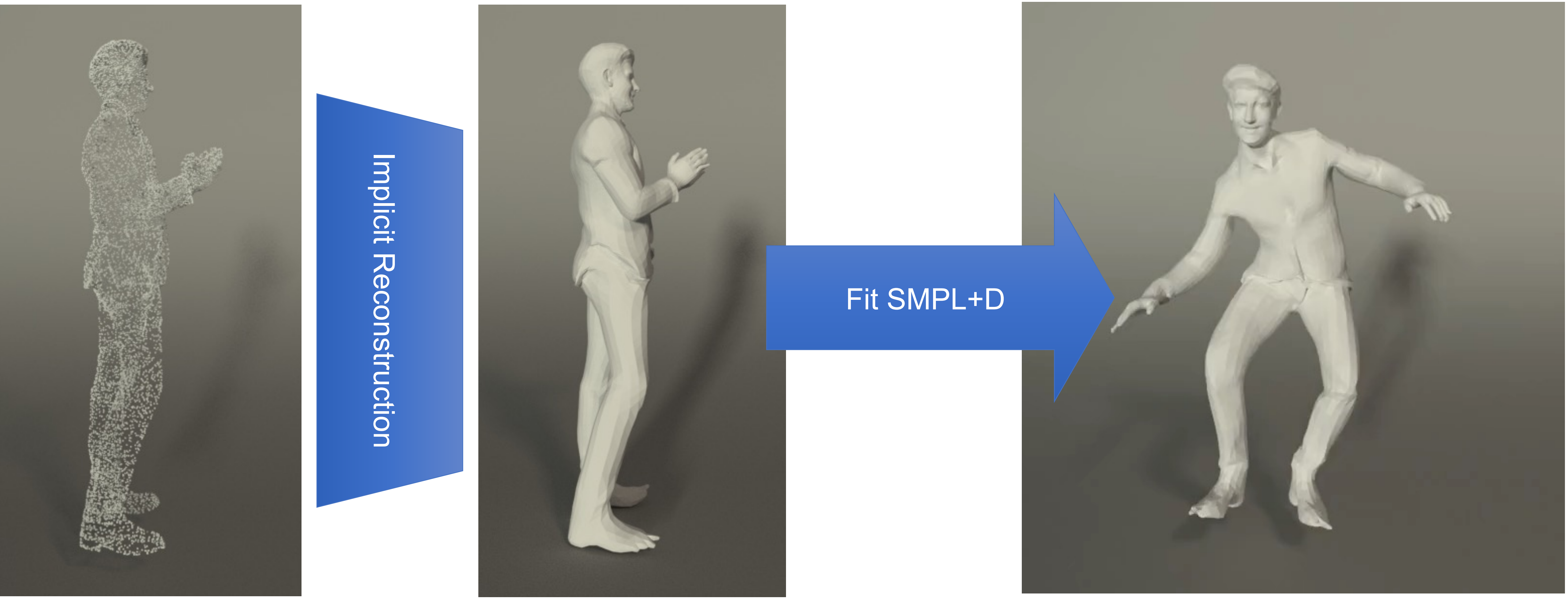
# Get detailed and controllable reconstructions.

Input PC Registration Input Motion Sequence Animated registration



IPNet. Bhatnagar et. al, ECCV'20

# IPNet: High level idea






IPNet. Bhatnagar et. al, ECCV'20

# Why combine implicit functions and parametric models?

**Implicit Reconstruction**

✓Better details.

✓Can handle arbitrary poses.

✗Just static meshes;
Can't do much.

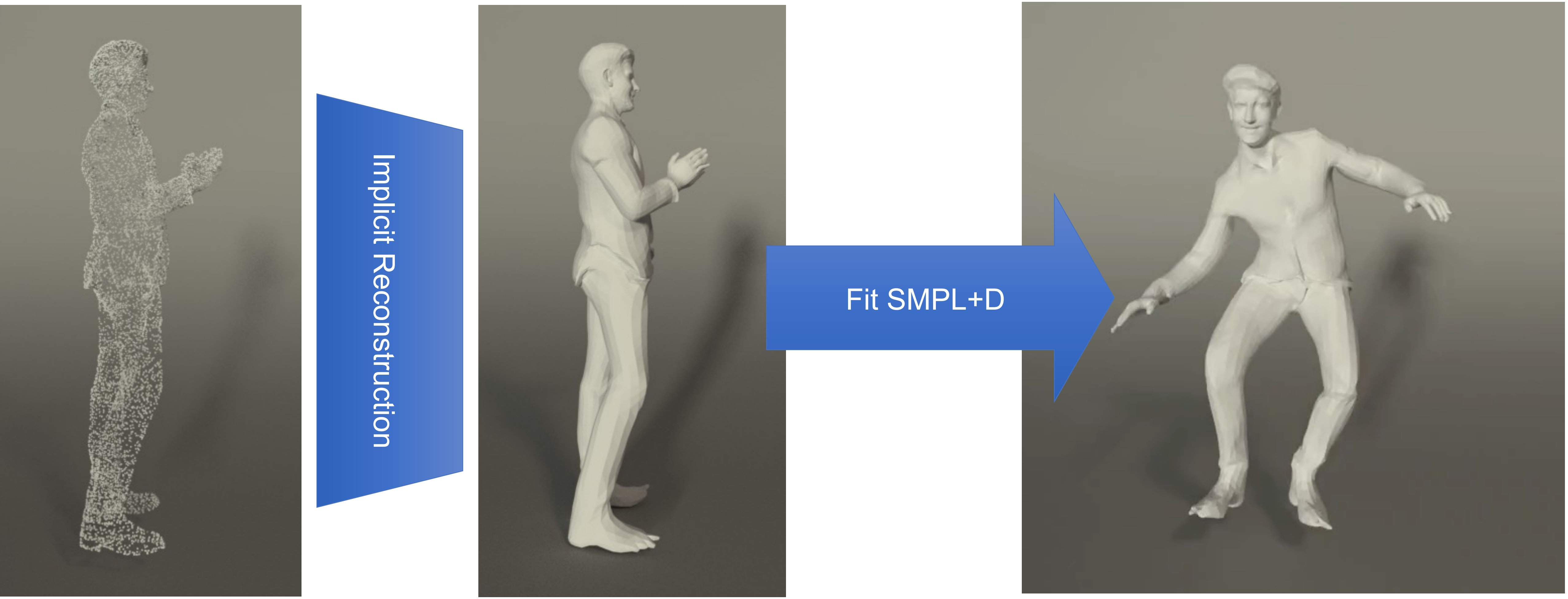
**Parametric Modelling**

✗Lacks details.

✗Generalization to complex poses is difficult.

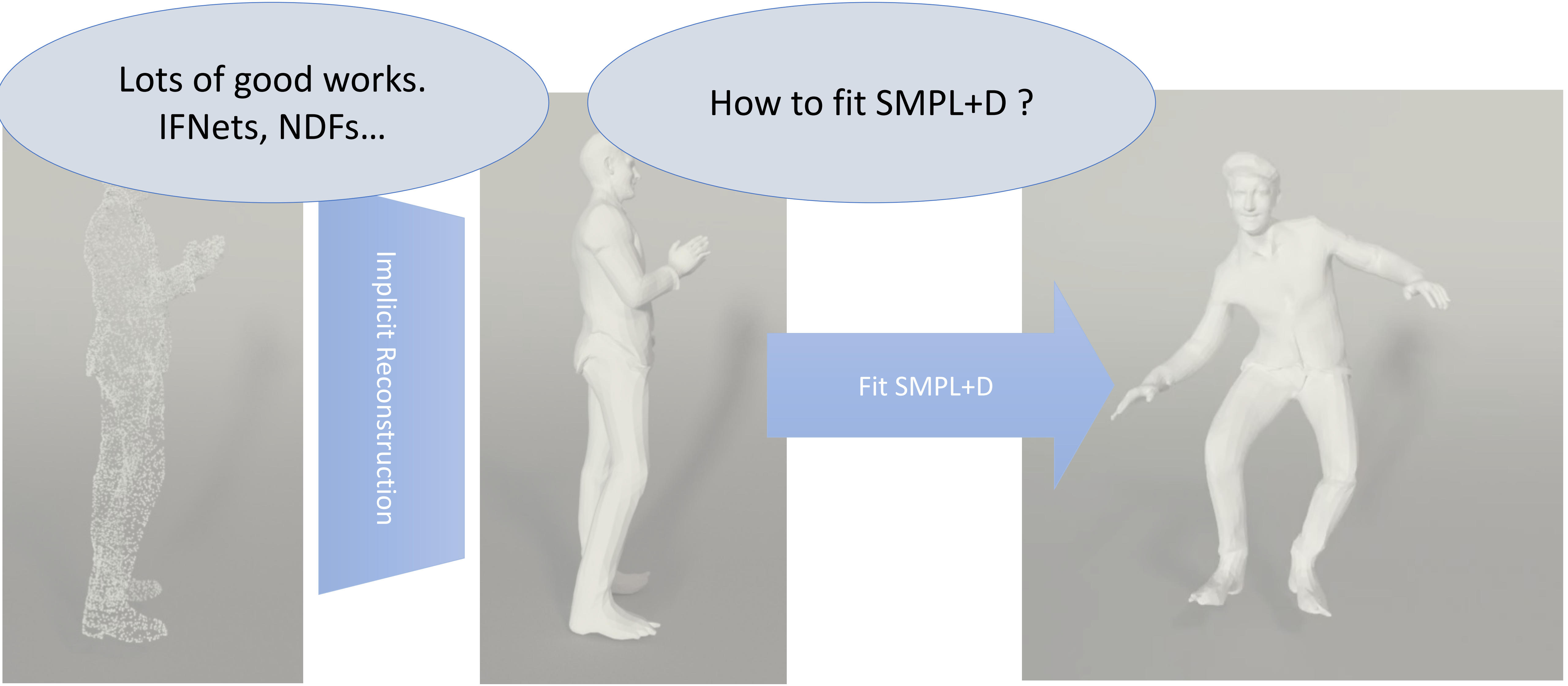
✓Can be re-shaped, re-posed etc.

# IPNet: High level idea



# Challenge

# How to fit SMPL+D? We saw ICP fail!



- **Problem**: ICP gets stuck due to bad correspondences and due to the fact that SMPL can not represent cloth, hair etc

- **Idea1**: Predict SMPL as an implicit surface to make fitting easy

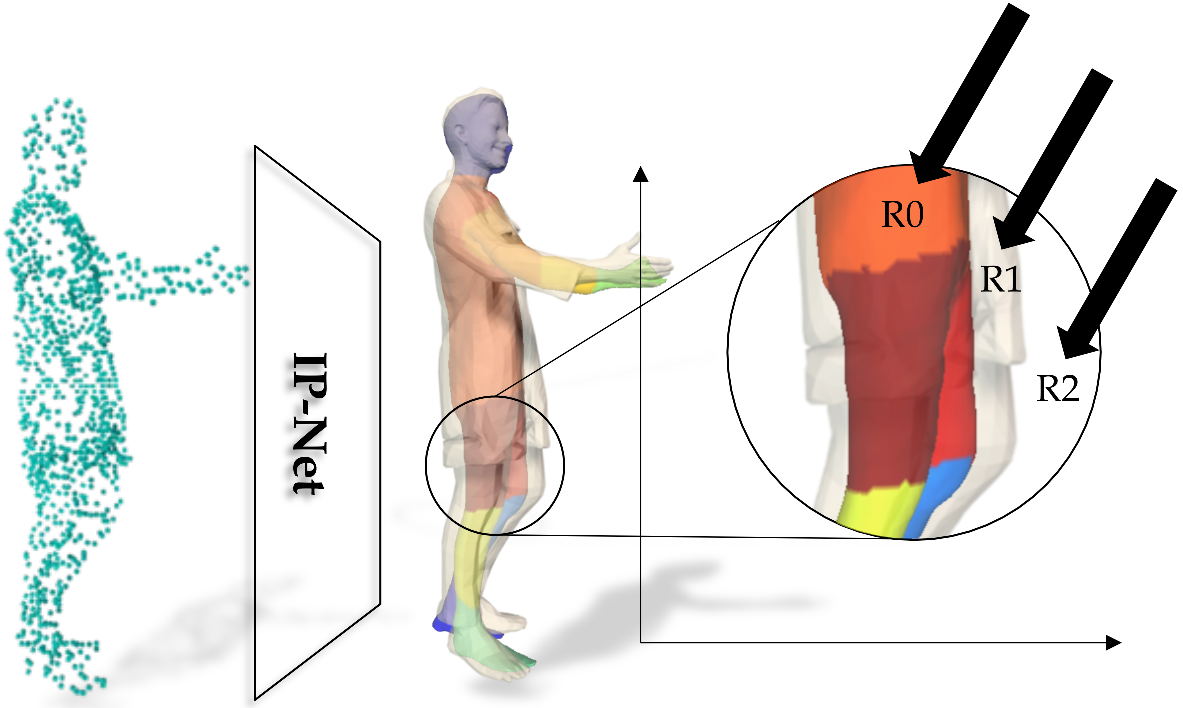
- **Idea2**: Learn to predict correspondences rather than using nearest point.

# IPNet: Predictions

- Double layer implicit function for outer and **inner** shape.

- Part correspondences to parametric model

$$f(\mathbf{p}|\mathcal{S}) \mapsto \{0,1,2\},\{1,...,N\}$$

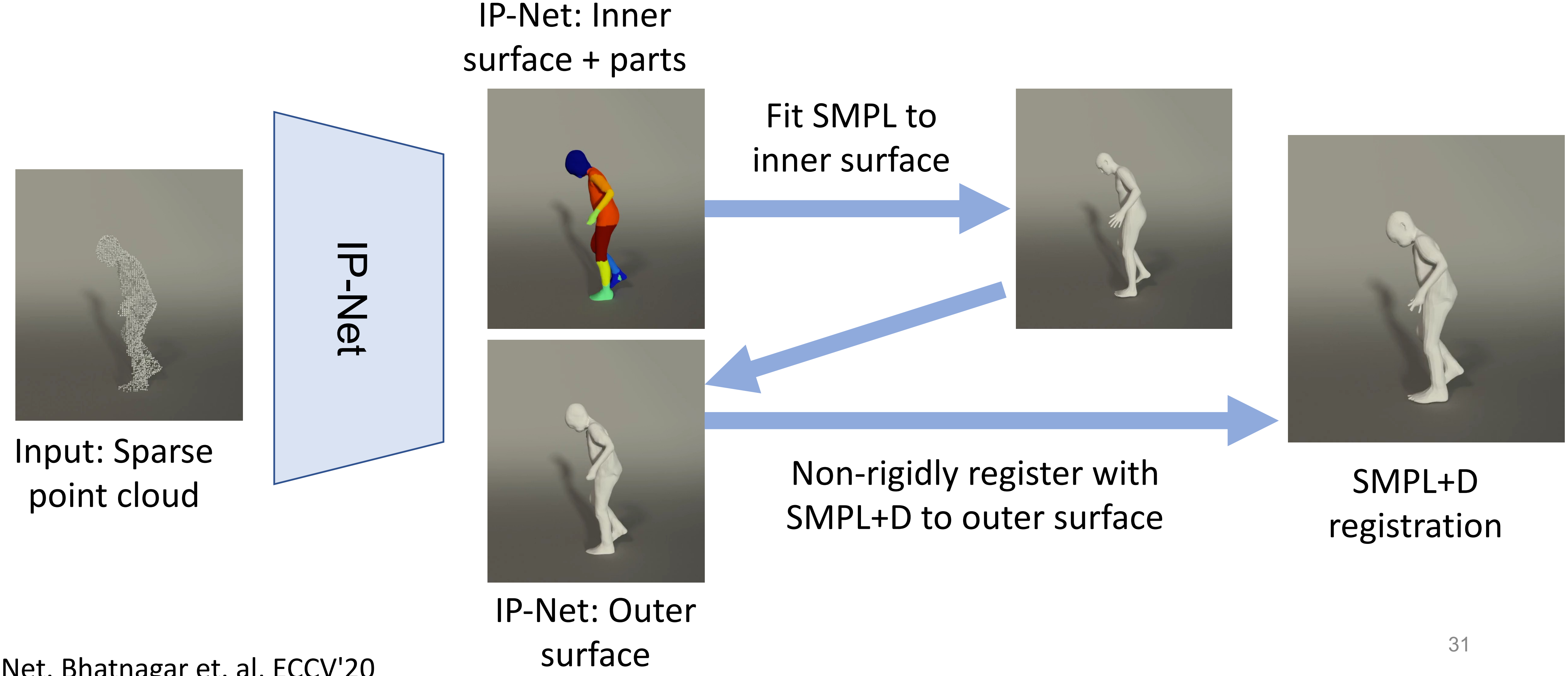





IPNet. Bhatnagar et. al, ECCV'20

# IPNet: Overview

**Implicit Reconstruction** | **Parametric Mesh**




IPNet. Bhatnagar et. al, ECCV'20

# Registering SMPL to IP-Net predictions

# Registering SMPL to IP-Net predictions

$$E(\underbrace{\boldsymbol{\theta}, \boldsymbol{\beta}, \mathbf{t}}_{\text{SMPL parameters}}) = w_{\text{data}} \underbrace{E_{\text{data}}}_{\substack{\text{Match SMPL surface to} \\ \text{the body surface} \\ \text{predicted by IP-Net}}} + w_{\text{part}} \underbrace{E_{\text{part}}}_{\substack{\text{Match parts on the} \\ \text{SMPL mesh to parts} \\ \text{predicted by IP-Net}}} + w_{\text{lap}} \underbrace{E_{\text{lap}}}_{\text{Laplacian regularizer}}$$

# Registering SMPL to IP-Net predictions

$$E_{\text{data}}(\boldsymbol{\theta}, \boldsymbol{\beta}, \mathbf{t}) = \frac{1}{|\mathcal{S}_{in}|} \sum_{\mathbf{v}_i \in \mathcal{S}_{in}} \underbrace{d(\mathbf{v}_i, \mathcal{M})}_{\text{Dist. from body to SMPL}} + w \cdot \frac{1}{|\mathcal{M}|} \sum_{\mathbf{v}_j \in \mathcal{M}} \underbrace{d(\mathbf{v}_j, \mathcal{S}_{in})}_{\text{Dist. from SMPL to body}}$$

$\mathcal{S}_{in}$ : body surface predicted by IP-Net

$\mathcal{M}$ : SMPL surface

$d(v, \mathcal{S})$ : distance of point $v$ from surface $\mathcal{S}$

# Registering SMPL to IP-Net predictions

$$E_{\text{part}}(\boldsymbol{\theta}, \boldsymbol{\beta}, \mathbf{t}) = \frac{1}{|\mathcal{S}_{in}|} \sum_{I=0}^{N-1} \sum_{\mathbf{v}_i \in \mathcal{S}_{in}} d(\mathbf{v}_i, \mathcal{M}^I)\delta(I^i = I)$$

Select body vertices corresponding to part *I*

Summation over SMPL parts

Dist. from body vertex to SMPL sub-mesh corresponding to part *I*

Summation over predicted body vertices

# Registering SMPL+D to IP-Net predictions

$$E_{\mathrm{data}}(\mathbf{D}, \boldsymbol{\theta}, \boldsymbol{\beta}, \mathbf{t}) = \frac{1}{|\mathcal{S}_o|} \sum_{\mathbf{v}_i \in \mathcal{S}_o} \underbrace{d(\mathbf{v}_i, \mathcal{M})}_{\text{Dist. from dressed surface to SMPL+D}} + w \cdot \frac{1}{|\mathcal{M}|} \sum_{\mathbf{v}_j \in \mathcal{M}} \underbrace{d(\mathbf{v}_j, \mathcal{S}_o)}_{\text{Dist. from SMPL+D to the dressed surface}}$$

$\mathbf{D}$ : per-vertex displacements on top of SMPL
$\mathcal{S}_o$ : dressed outer surface predicted by IP-Net
$\mathcal{M}$ : SMPL+D surface
$d(v, \mathcal{S})$ : distance of point $v$ from surface $\mathcal{S}$

# IPNet: Results

# Single View Point Cloud Registration

Input: Single View PC | IP-Net inner surface & parts | IP-Net outer surface | Registration

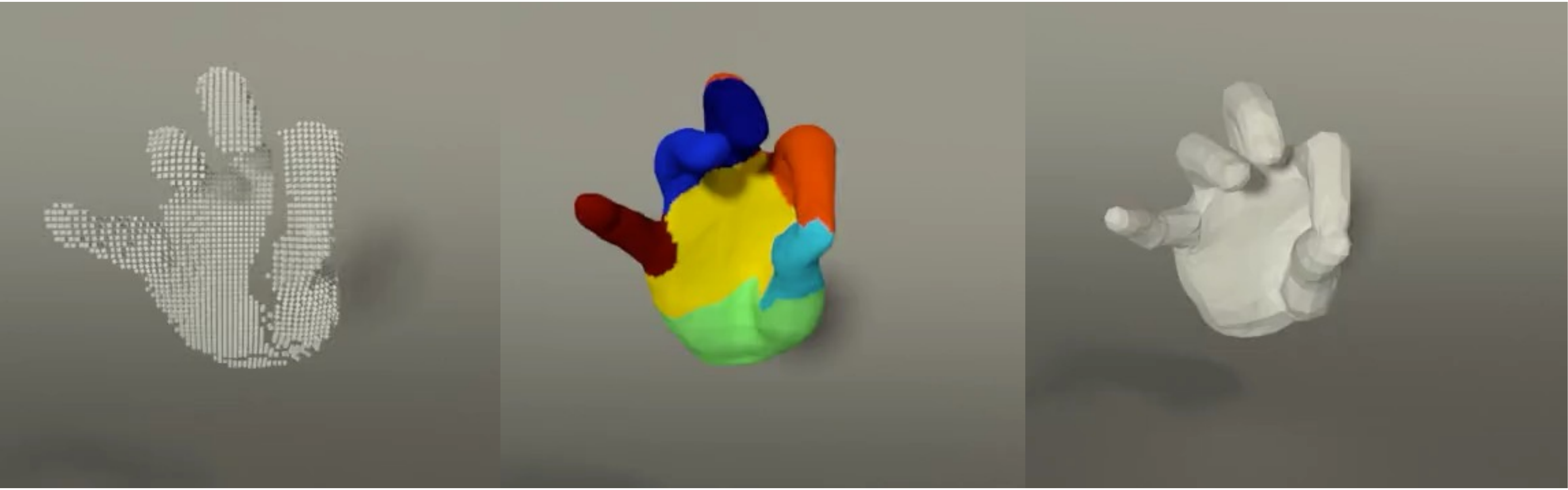
# We can animate our reconstructions

Input: Dense PC　　Registration　　Input: Motion sequence　　Animated registration



IPNet. Bhatnagar et. al, ECCV'20

# IPNet generalises to other domains.



Input: Single View PC | IP-Net surface & parts | Registration

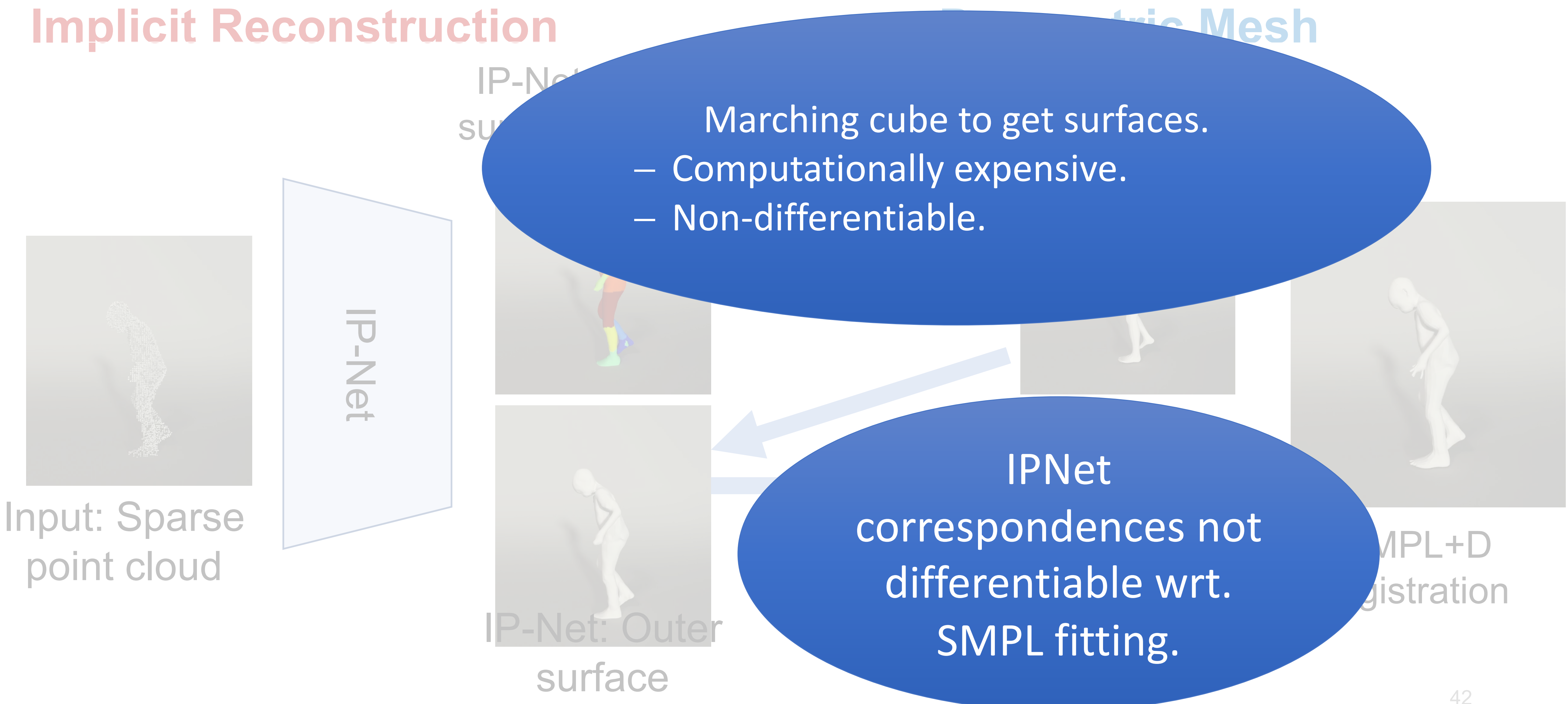


IPNet. Bhatnagar et. al, ECCV'20

# What does "learning" bring over ICP?

- Learnt correspondences more reliable than just nearest point.
- We can learn to complete/ denoise input shape.
  ICP struggles with with partial data.

# IPNet: Limitations




Marching cube to get surfaces.

- Computationally expensive.
- Non-differentiable.

IPNet correspondences not differentiable wrt. SMPL fitting.

# Q. Is ICP differentiable wrt. SMPL fitting?

- Recall ICP formulation…
- Is it differentiable?

## Iterative Closest Point (ICP)

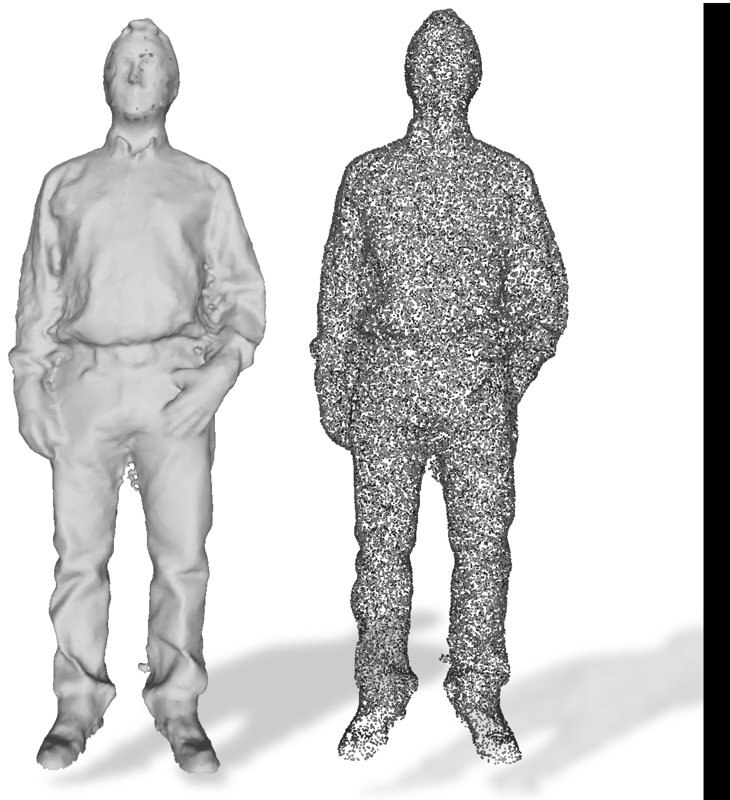
1. initialise $$f^0 = \{\mathbf{R} = \mathbf{I}, \mathbf{t} = \frac{\sum \mathbf{y}_i}{N} - \frac{\sum \mathbf{x}_i}{N}, s = 1\}$$
2. compute correspondences according to current best transform $$\mathbf{x}_i^{j+1} = \arg\min_{\mathbf{x} \in \mathbf{X}} \|f^j(\mathbf{x}) - \mathbf{y}_i\|^2$$
3. compute optimal transformation ( $\mathbf{s}, \mathbf{R}, \mathbf{t}$ )with Procrustes $$f^{j+1} = \arg\min_f \sum_i \|f(\mathbf{x}_i^{j+1}) - \mathbf{y}_i\|^2$$

- Can we make correspondences differentiable?
  → **End-to-end differentiable registration?**

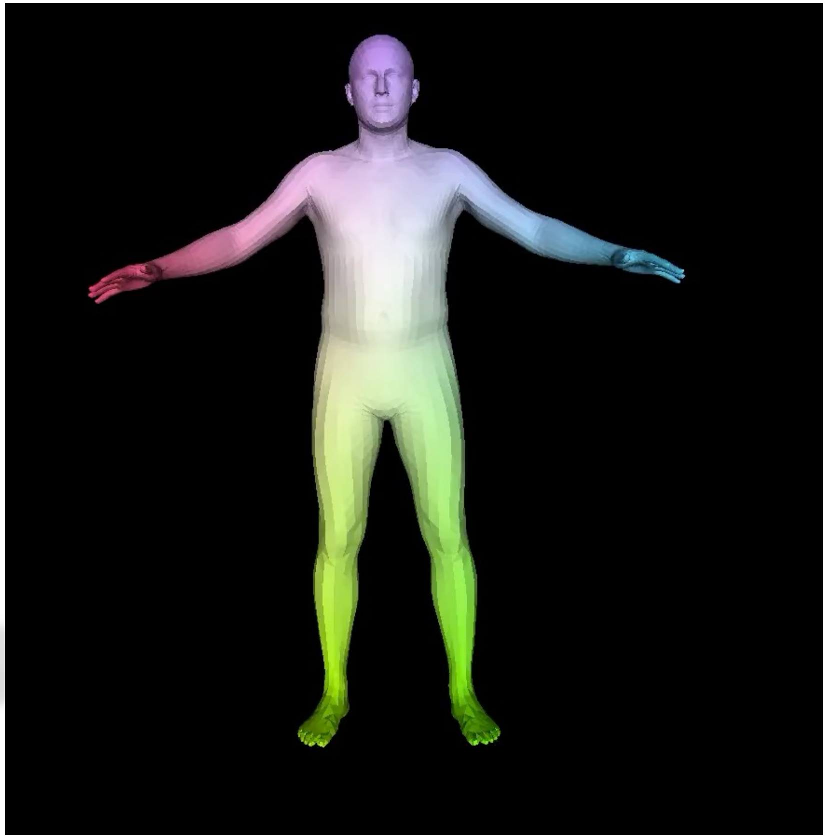
- Can we remove expensive marching cubes?

# 3D scan → Human Model

Input: 3D scan/ Pointcloud



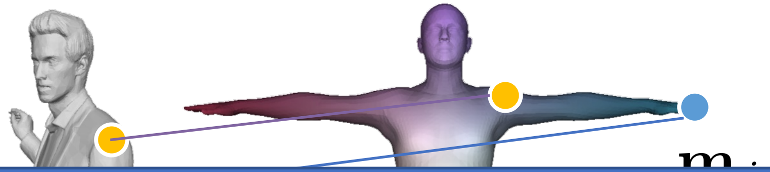
Colour coded SMPL model



Output: Registered SMPL+D

# Problem: Traditional registration

1. Get correspondences.
   - Keypoints / Landmarks



- Instance specific
- Prone to local minima
- Not End-to-end Differentiable wrt. Correspondences !!

   - Optimize the model parameters.

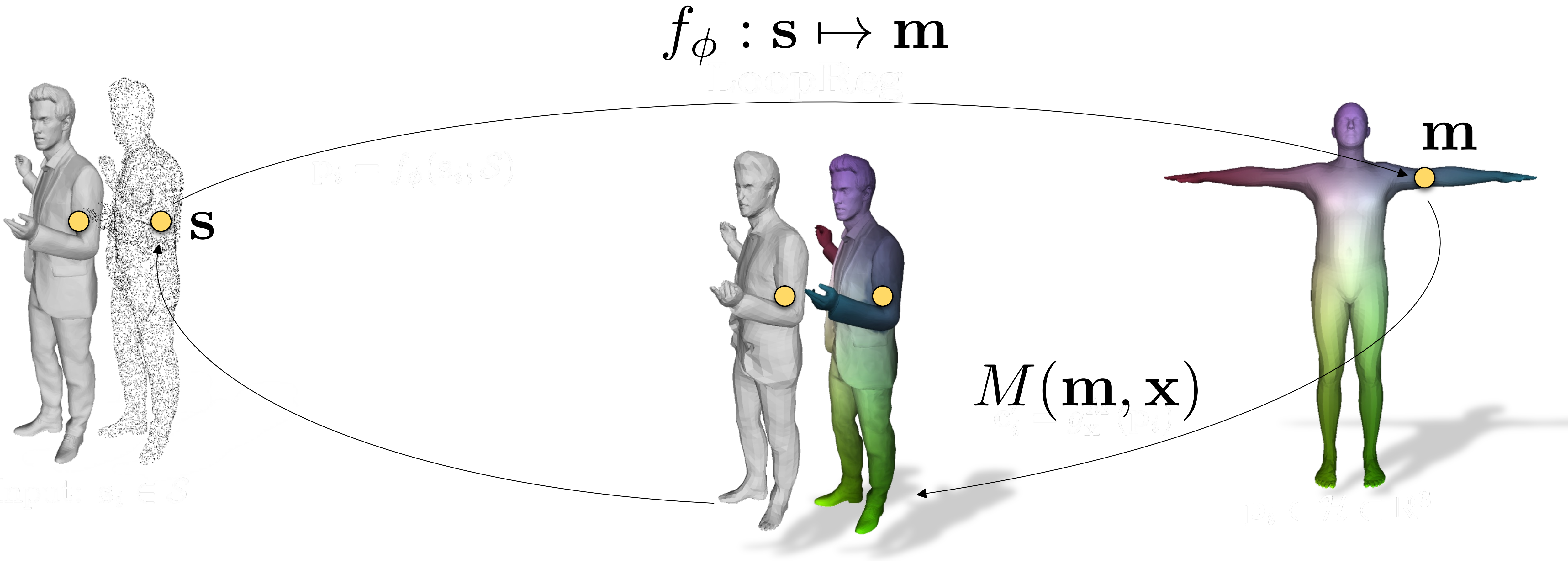
$$\arg\min_{\mathcal{C},\mathbf{x}} \sum_{\mathbf{s},\mathbf{m}\in\mathcal{C}} \|\mathbf{s}_i - M(\mathbf{m}_j, \mathbf{x})\|^2$$

3. Iterate over 1 & 2.



LoopReg. Bhatnagar et. al, NeurIPS'20

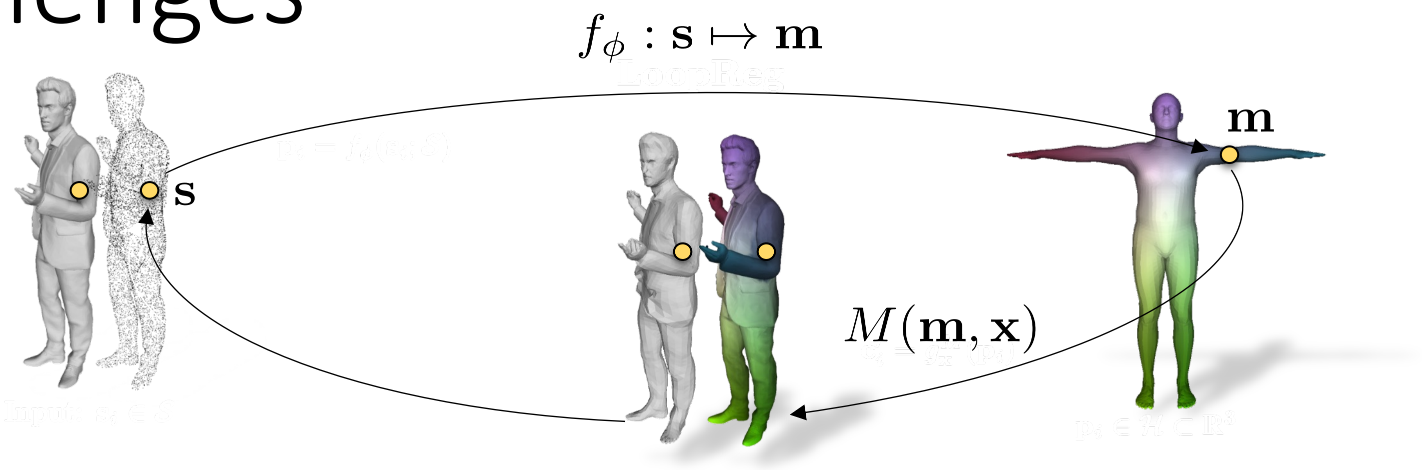
# Can we jointly optimize over model and correspondences without supervision?






LoopReg. Bhatnagar et. al, NeurIPS'20

# Key challenges




1. Can we jointly train the network $f_\phi$ and optimize $\mathbf{x}$ without supervision?

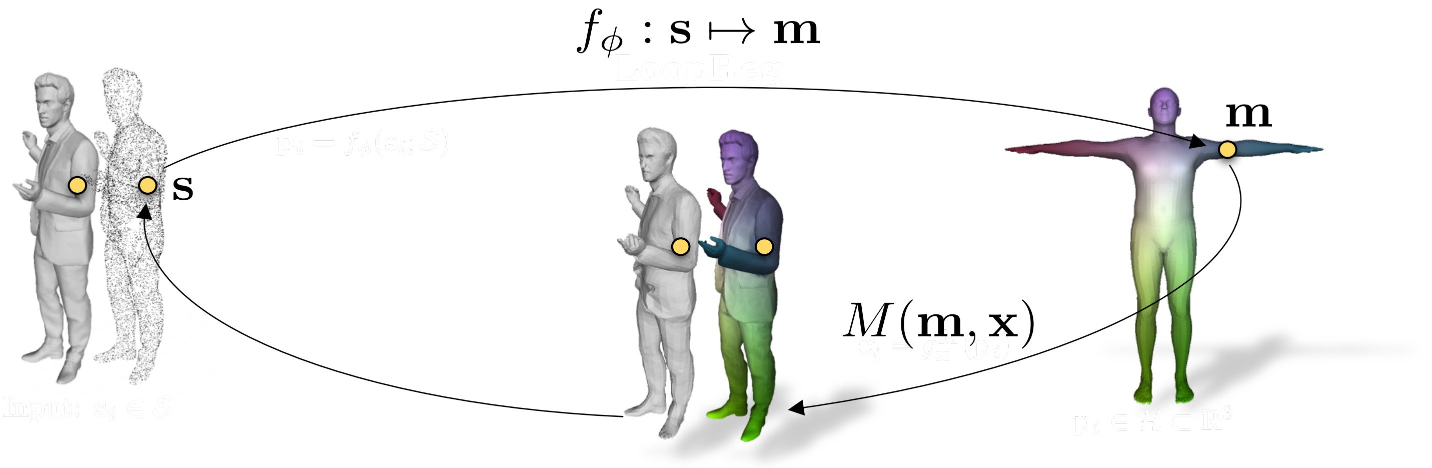
2. How to ensure that correspondence predictions lie on the model surface?

3. Integrate correspondence prediction with model fitting.



LoopReg. Bhatnagar et. al, NeurIPS'20

# Can we jointly optimize over model and correspondences without supervision?

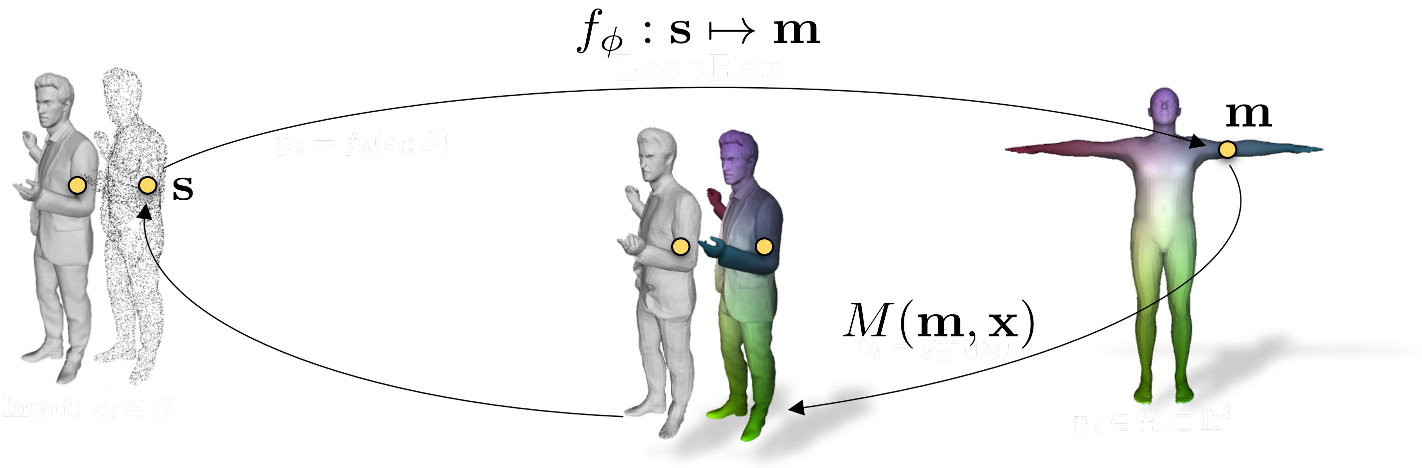



$$L_{\text{self}}(\phi, \mathcal{X}) = \sum_{j=1}^{N} \sum_{\mathbf{s}_i \in \mathcal{S}_j} \operatorname{dist}(\mathbf{s}_i, M(\mathbf{m}_k, \mathbf{x}_j))$$



LoopReg. Bhatnagar et. al, NeurIPS'20

# Let a Neural Network predict the correspondences.



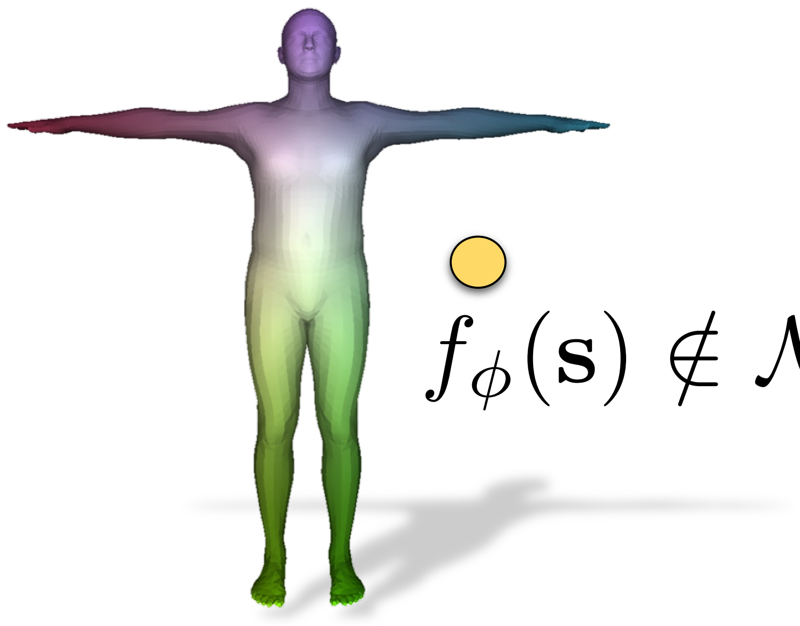

$$L_{\text{self}}(\phi, \mathcal{X}) = \sum_{j=1}^{N} \sum_{\mathbf{s}_i \in \mathcal{S}_j} \text{dist}(\mathbf{s}_i, M(f_\phi(\mathbf{s}), \mathbf{x}_j))$$



LoopReg. Bhatnagar et. al, NeurIPS'20

# NN predicted correspondences don't lie on the model surface.

Why not learn directly? $f_\phi : \mathbf{s} \mapsto \mathbf{m}$



$$f_\phi(\mathbf{s}) \notin \mathcal{M}$$

Deformation model (SMPL) only defined for surface points on the manifold

$$\mathcal{M}$$

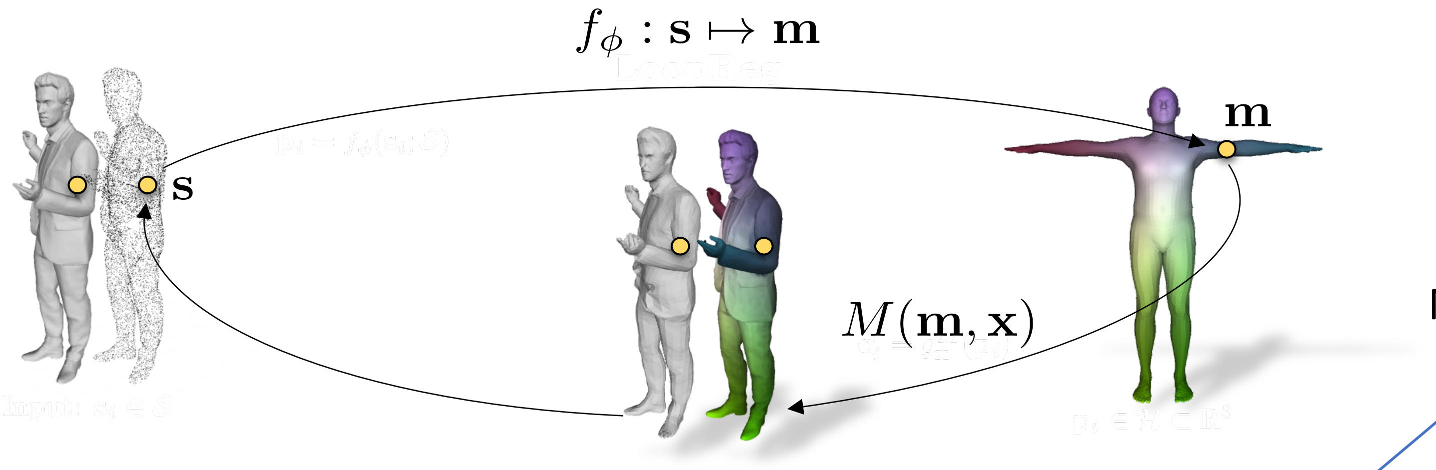
$$M(\mathbf{m}, \mathbf{x}) : \mathbf{m} \in \mathcal{M} \mapsto \mathbf{m}' \in \mathbb{R}^3$$

Not defined for off-manifold

$$M(f_\phi(\mathbf{s}))??$$

LoopReg. Bhatnagar et. al, NeurIPS'20

# NN predicted correspondences don't lie on the model surface.

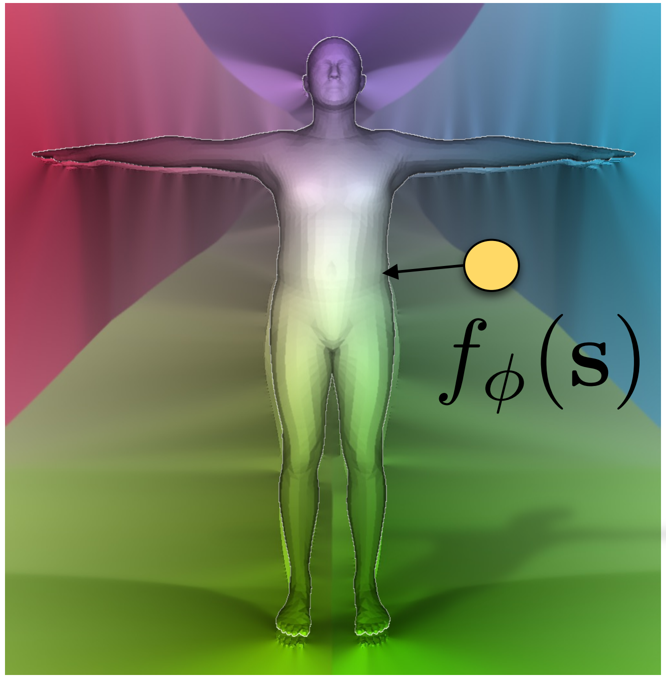



Not defined!!

$$L_{\text{self}}(\phi, \mathcal{X}) = \sum_{j=1}^{N} \sum_{\mathbf{s}_i \in \mathcal{S}_j} \text{dist}(\mathbf{s}_i, \boxed{M(f_\phi(\mathbf{s})}, \mathbf{x}_j))$$

LoopReg. Bhatnagar et. al, NeurIPS'20

# How to ensure that NN predicted correspondences lie on the model surface?



$$f_\phi(\mathbf{s}) \notin \mathcal{M}$$

DISTANCE TRANSFORM
BASED DIFFUSION

1) Diffuse the SMPL model beyond the surface

$$M(\mathbf{m}, \mathbf{x}) : \mathbf{m} \in \mathcal{M} \mapsto \mathbf{m}' \in \mathbb{R}^3$$

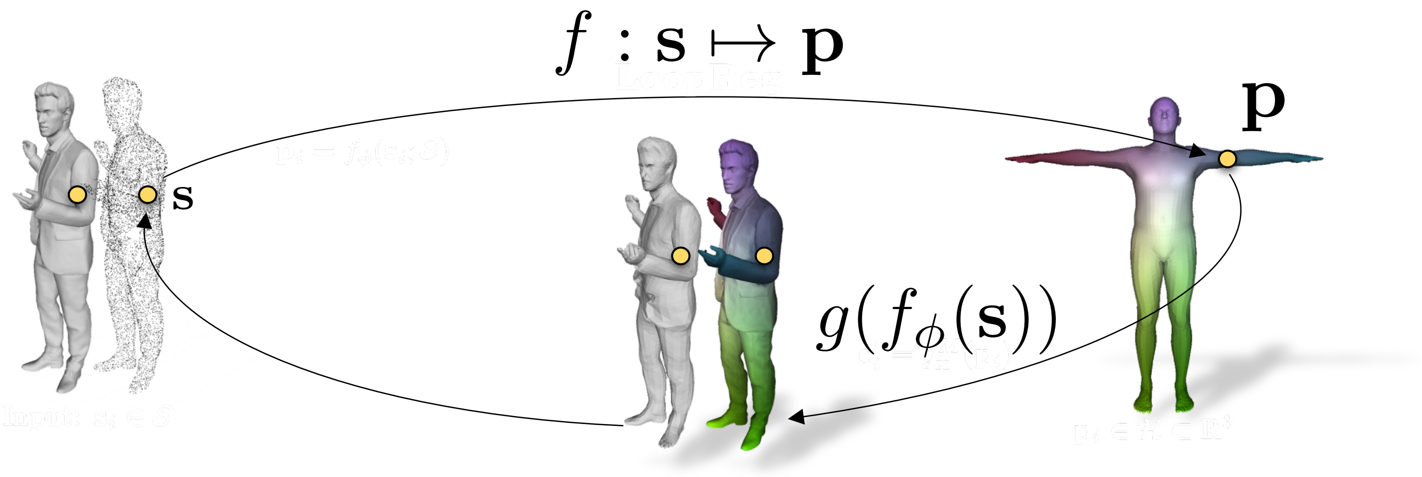
$$g(\mathbf{p}, \mathbf{x}) : \mathbf{p} \in \mathcal{H} \subset \mathbb{R}^3 \mapsto \mathbf{p}'$$

2) Add a Lagrangian constraint to force predictions to lie on the manifold

$$L_{\text{surface}} = \text{dist}_{\mathcal{M}}(f_\phi(\mathbf{s}))$$

LoopReg. Bhatnagar et. al, NeurIPS'20

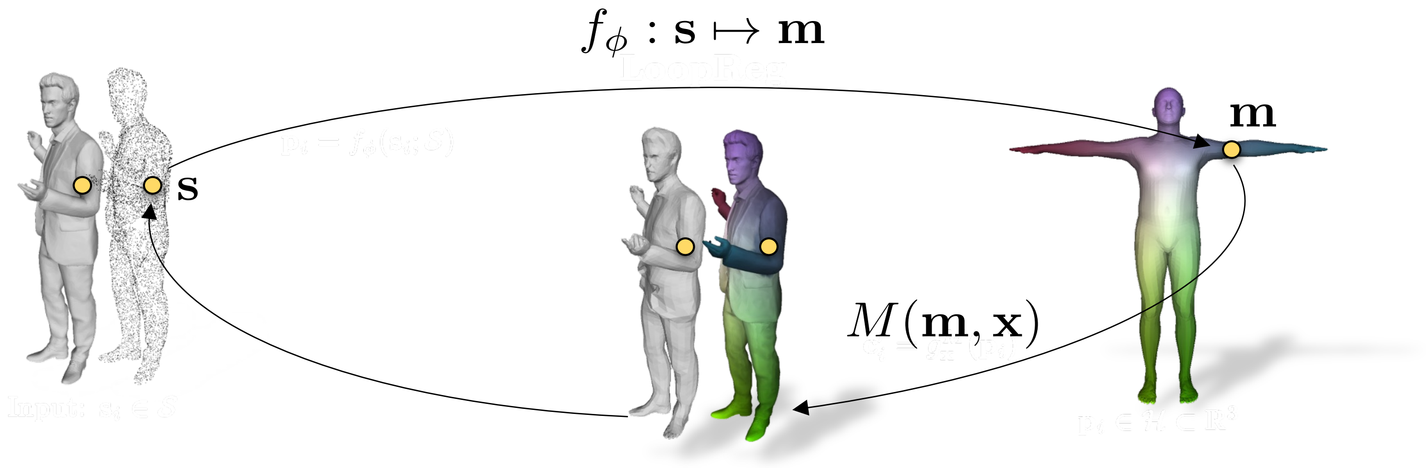
# Use diffused SMPL to get valid function in $\mathbb{R}^3$




$$L_{\text{self}}(\phi, \mathcal{X}) = \sum_{j=1}^{N} \sum_{\mathbf{s}_i \in \mathcal{S}_j} \text{dist}(\mathbf{s}_i, \boxed{g(f_\phi(\mathbf{s}))}, \mathbf{x}_j)$$

LoopReg. Bhatnagar et. al, NeurIPS'20

# We can jointly optimize over model and correspondences without supervision.




$$L_{\text{self}}(\phi, \mathcal{X}) = \sum_{j=1}^{N} \sum_{\mathbf{s}_i \in \mathcal{S}_j} \text{dist}(\mathbf{s}_i, \boxed{g(f_\phi(\mathbf{s}))}, \mathbf{x}_j) + \boxed{\lambda \cdot \text{dist}_{\mathcal{M}}(f_\phi(\mathbf{s}))}$$

LoopReg. Bhatnagar et. al, NeurIPS'20

# Performance improves with more unlabelled data

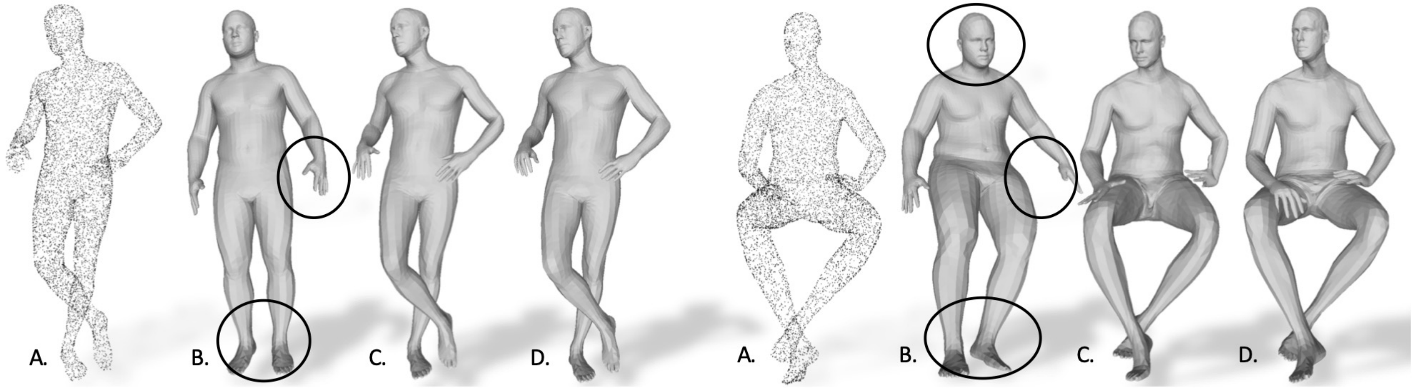
| **Unsupervised %** | **0%** | **10%** | **25%** | **50%** | **75%** | **100%** |
|---|---|---|---|---|---|---|
| (a) v2v (cm) | 9.3 | 8.4 | 6.3 | 4.1 | 2.7 | 1.5 |
| (b) s2s (mm) | 6.8 | 6.6 | 6.2 | 5.5 | 5.1 | 4.2 |

Table 2: Performance of the proposed approach increases as we add more unsupervised data for training. Here 100% corresponds to 2631 scans. Out of the 2631 scans 1000 were also used for supervised warm-start. We report vertex-to-vertex (v2v) and bi-directional surface-to-surface (s2s) errors and clearly show that adding more unsupervised data improves registration performance.



LoopReg. Bhatnagar et. al, NeurIPS'20

# Comparison to competing approaches




A) Input, B) Alldieck et al. CVPR'19 C) Ours D) Ground Truth

| **Method** | **Inter-class AE** (cm) | **Intra-class AE** (cm) |
|---|---|---|
| FMNet [52] | 4.83 | 2.44 |
| FARM [49] | 4.12 | 2.81 |
| LBS-AE [44] | 4.08 | 2.16 |
| 3D-CODED [32] | 2.87 | 1.98 |
| **Ours** | **2.66** | **1.34** |

Results on FAUST correspondence prediction challenge.

# Summary

- ICP is simple conceptually, but finding closest points is prone to local minima
- IPNet combines learned implicit surface reconstruction and model fitting
  - Predict double layer surface (inner and outer) with part correspondences
  - Fit SMPL to inner layer and expand to outer layer
- LoopReg makes registration differentiable wrt. correspondence prediction.