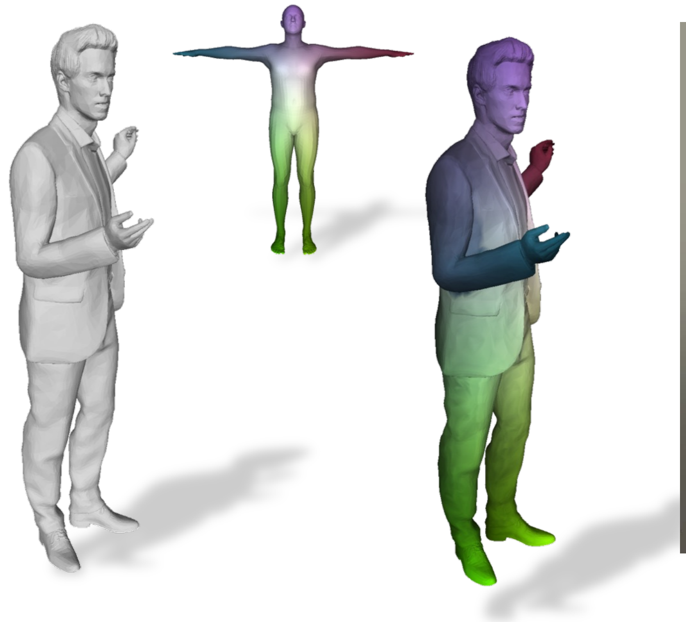
Virtual Humans – Winter 23/24

Lecture 6_1 – Fitting SMPL to Images with Optimization

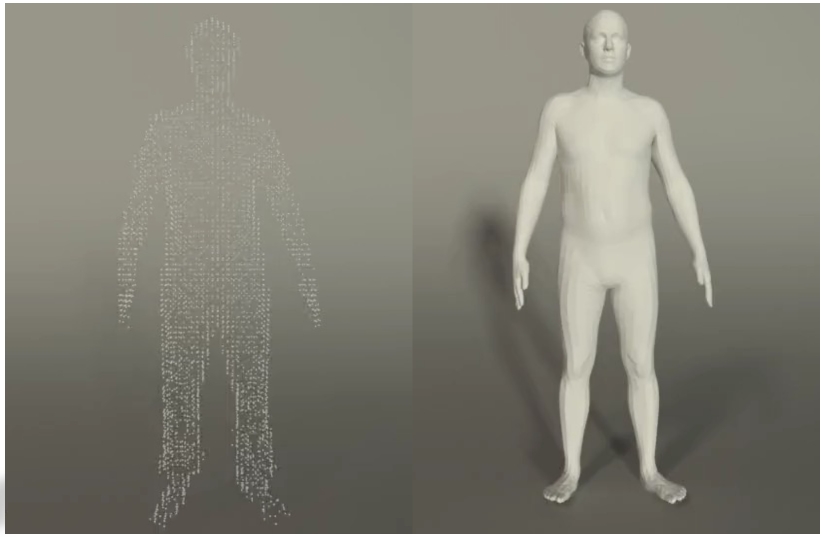
Prof. Dr.-Ing. Gerard Pons-Moll
University of Tübingen / MPI-Informatics

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

We have seen how to fit SMPL to scans



Correspondences



Tracking

Animation / Control

How can we infer 3D human pose and shape from a single image?

Understand people in images

Estimate 3D shape and pose

"See" the person in 3D

Two ways to estimate 3D humans

- Discriminative Models
  - *"condition on the image"*

- Generative Models
  - *"explain the image"*

Why is it Hard?

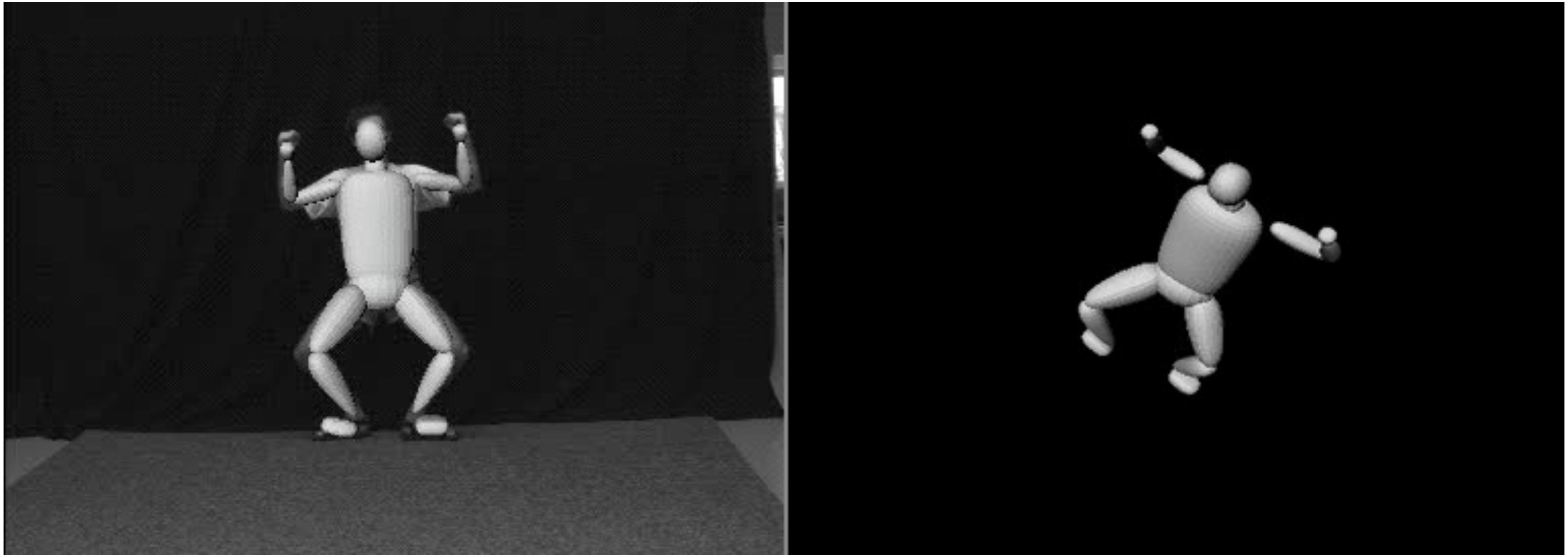
Why is it Hard?

- Many degrees of freedom.
- Highly Dynamic / Skinning/ Clothing / Outdoor.
- Large variability and individuality of Motion patterns.

http://www.google.de/images

Why is it hard?

- Depth ambiguity: many 3D poses produce the exact same projection!



Sminchisescu and Triggs. CVPR'01

Can we use prior information about humans?

- We know humans have a fixed skeletal structure.

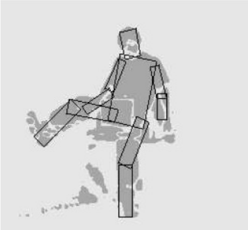
- Motion is mostly articulated.

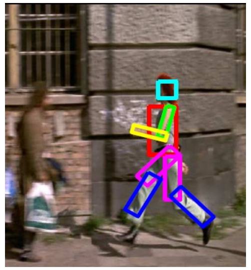
- Human shape is roughly symmetric, and lives in a subspace.

We need a body model

Body models in the past

2D

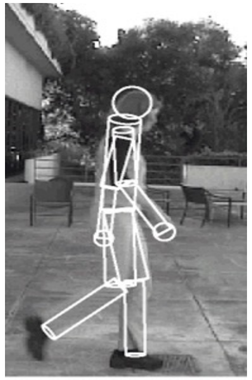


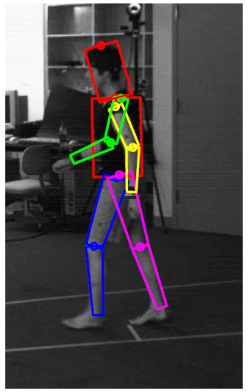


Felzenszwalb et.al
Ramanan et.al.
Andriluka et.al.

3D

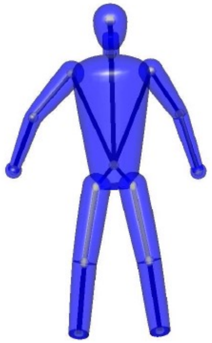
Cylinders

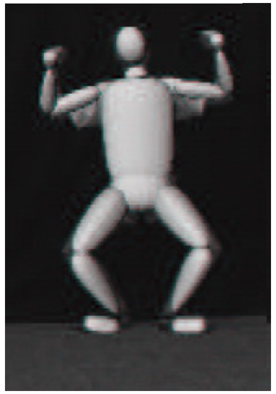




Kjellström et.al.
Sigal et.al.

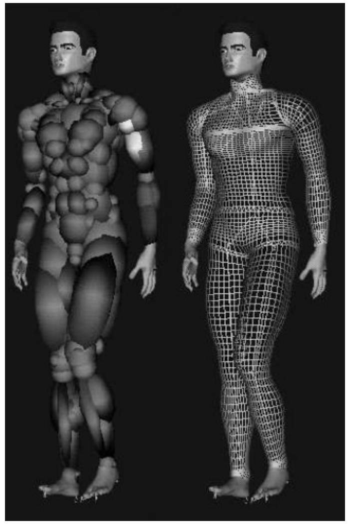
Ellipsoids





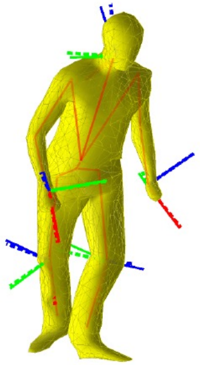
Kehl and Van Gool
Sminchisescu and Triggs

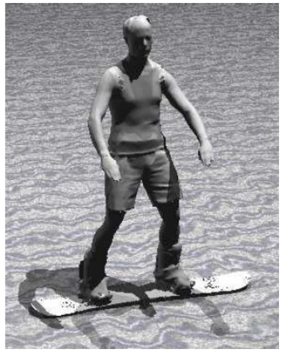
Gaussian Blobs

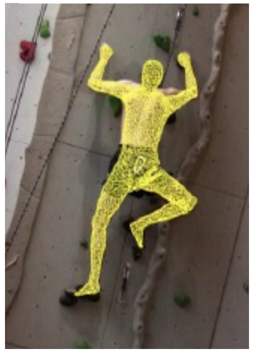


Plaenkers and Fua

Rigged scan







Pons-Moll et.al.
Rosehnahn et.al.
Gall et.al.

Nowadays

Nowadays **SMPL** is the de-facto model for human **pose and shape** estimation from **images**.

Problem formulation

- Input: RGB image
- Estimate: Model (SMPL) parameters
  - Pose
  - Shape
  - Camera (optional)

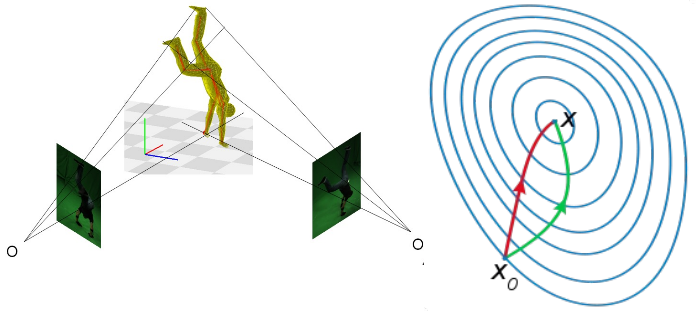
Inference with a generative model eg. SMPL

Posterior		Likelihood		Prior
$p(\mathbf{x}\|\mathbf{I})$	$\propto$	$p(\mathbf{I}\|\mathbf{x})$	$\times$	$p(\mathbf{x})$

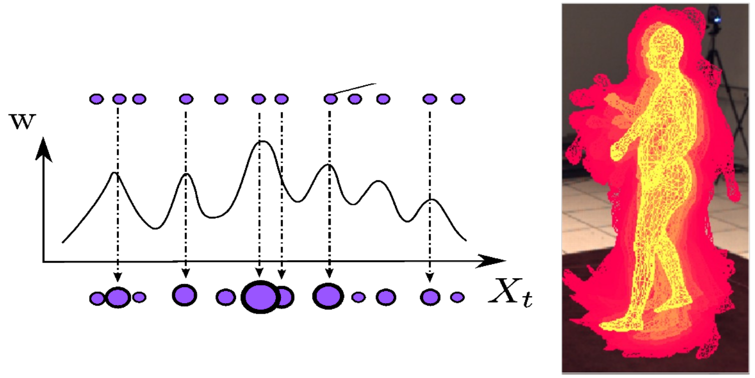
Model params.
(eg. θ, β)

Input/ Observation
(eg. RGB image)

How to model $p(\mathbf{x}|\mathbf{I})$?



Optimization



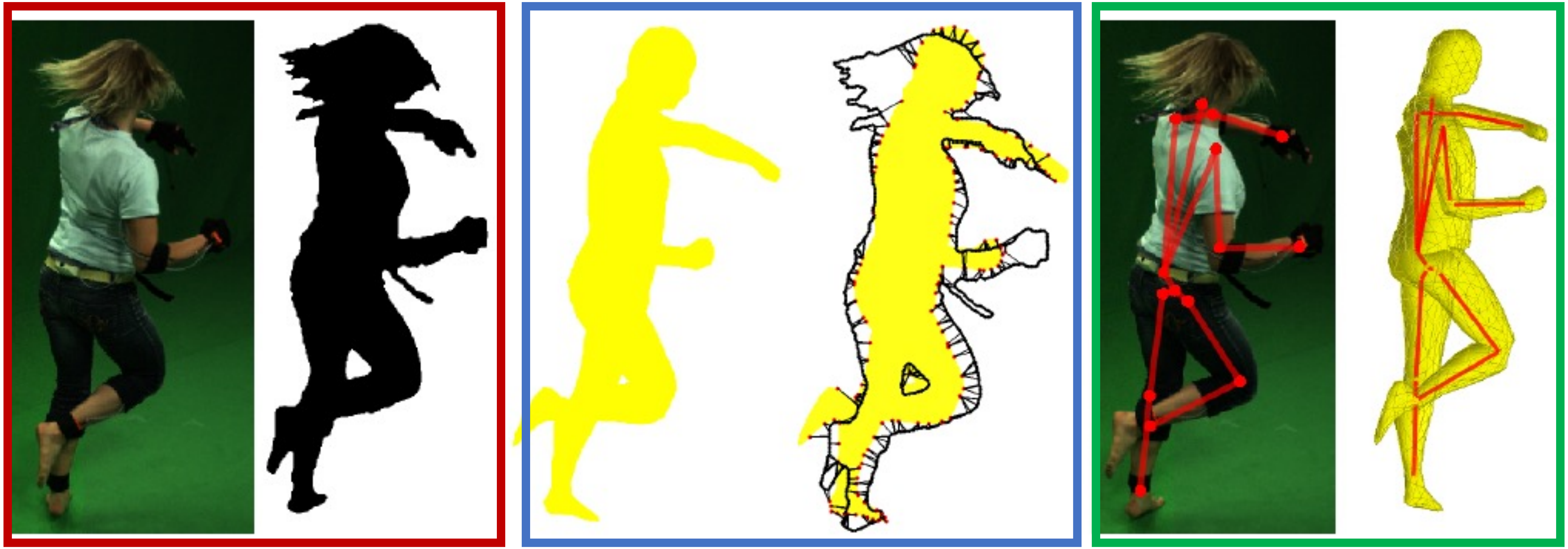
Bayesian models

Map of $p(\mathbf{x}|\mathbf{I})$

Approx. $p(\mathbf{x}|\mathbf{I})$ with weighted samples

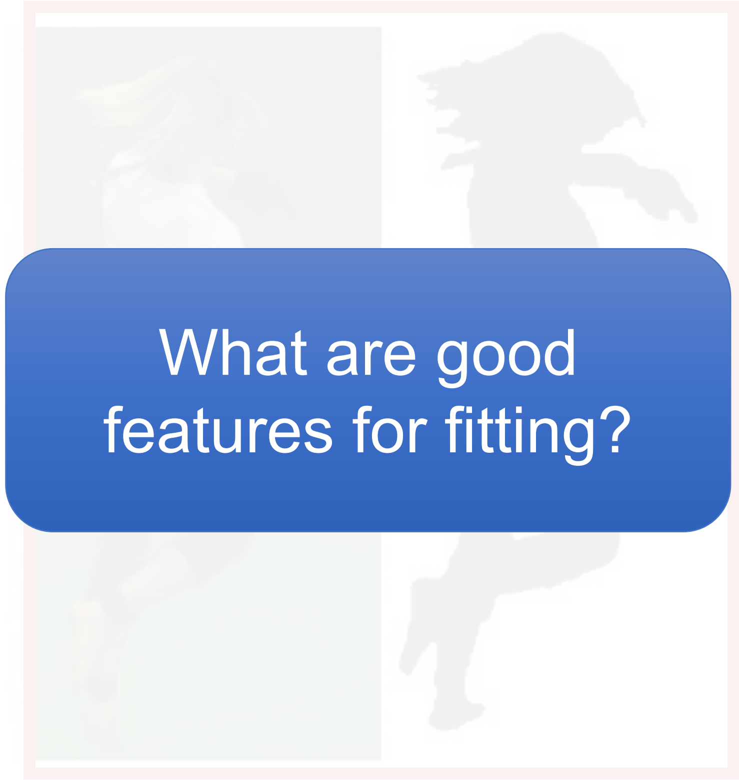
General framework for optimization

1. Extract features
2. Predict and match
3. Optimize

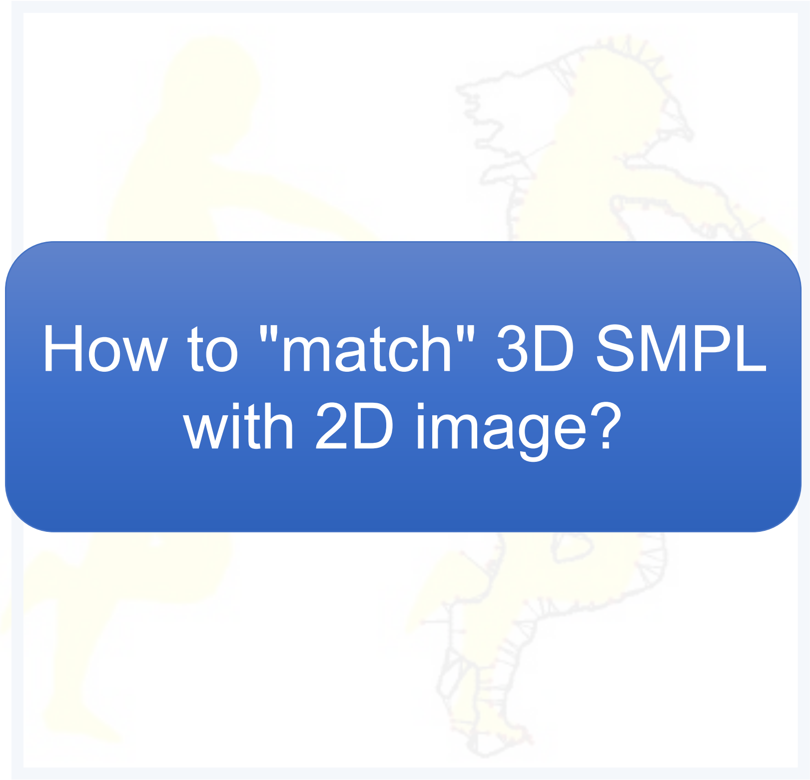


General framework for optimization

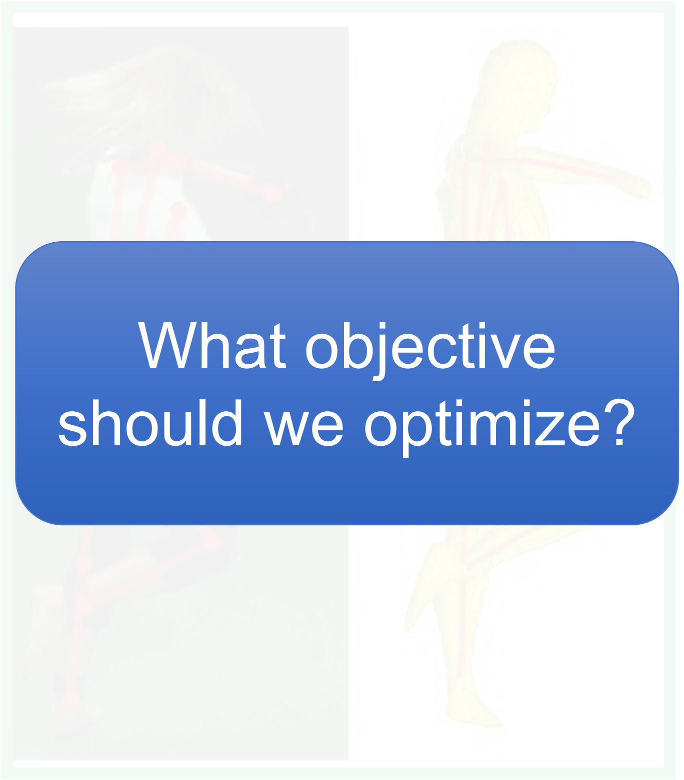
1. Extract features
2. Predict and match
3. Optimize



What are good features for fitting?



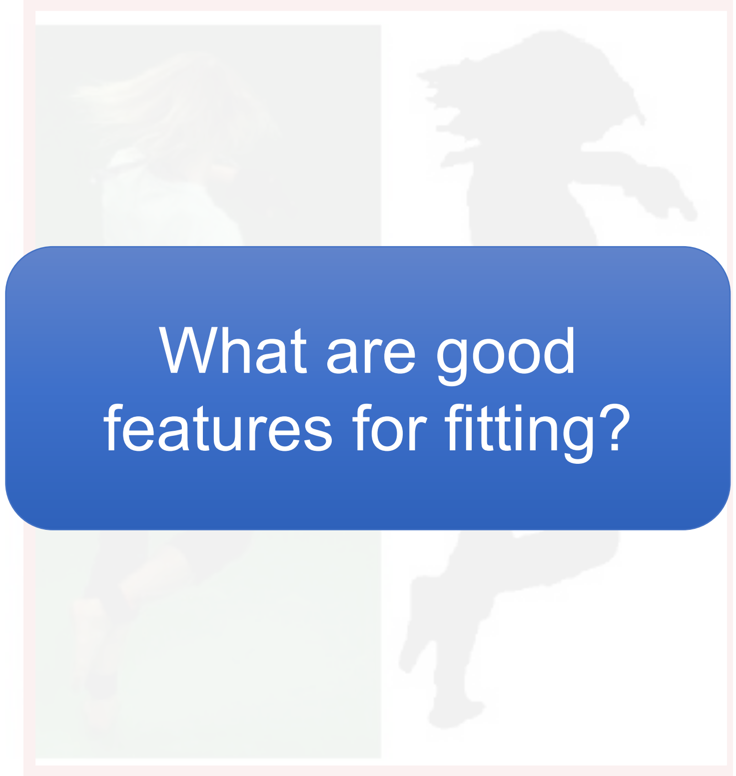
How to "match" 3D SMPL with 2D image?



What objective should we optimize?

General framework for optimization

1. Extract features
2. Predict and match
3. Optimize



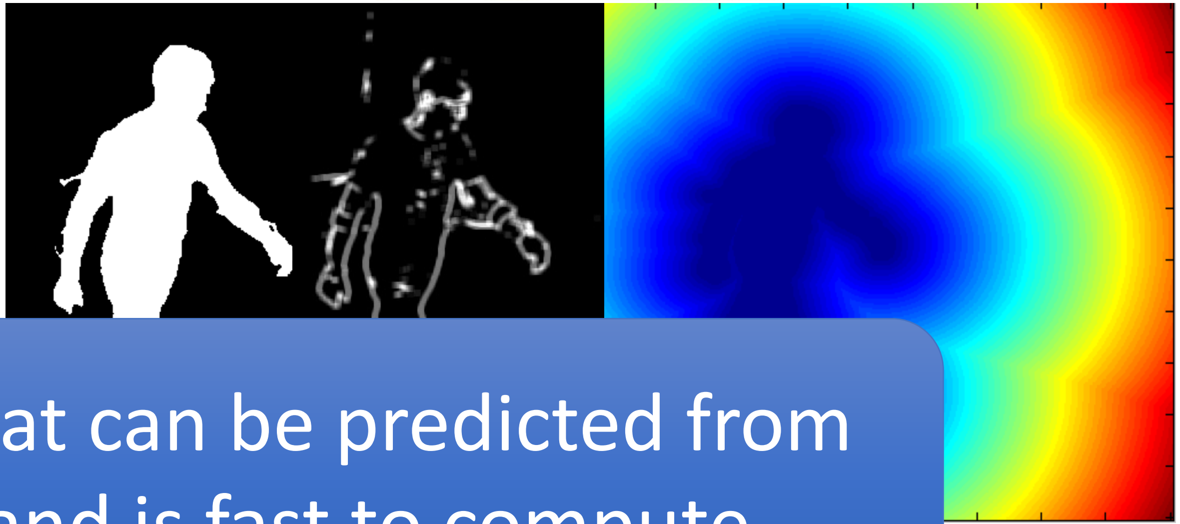
What are good features for fitting?

How to "match" 3D SMPL with 2D image?

What objective should we optimize?

What are good features for fitting?

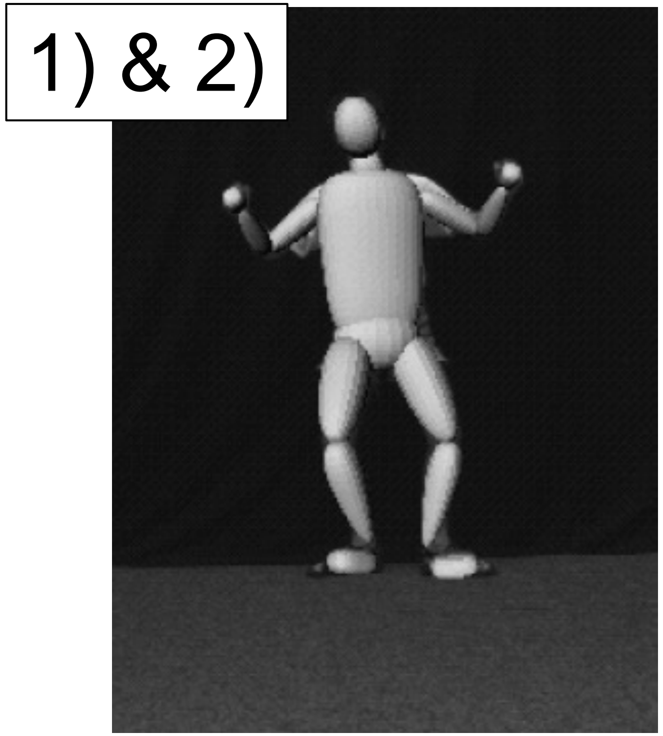
- Silhouettes
- Edges
- Distance transforms
- SIFT
- Optic flow
- Appearance
- …



Any feature that can be predicted from the model and is fast to compute

Let's look at Distance Transform for example

Inconsistent



Consistent

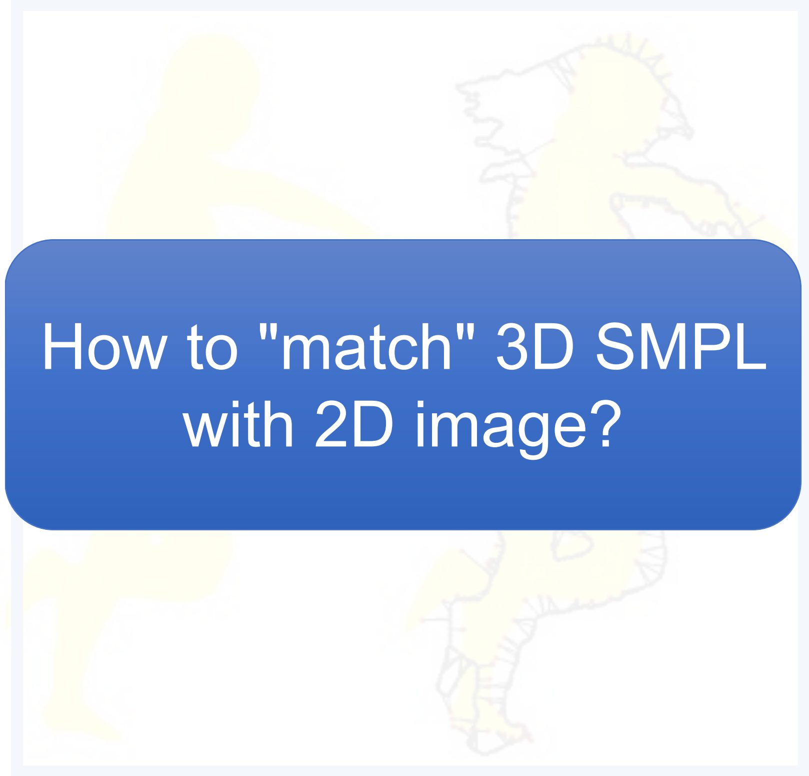
1) Push model inside silhouette
2) Force the model to explain the image

Sminchisescu F & G 2001

General framework for optimization

1. Extract features
2. Predict and match
3. Optimize

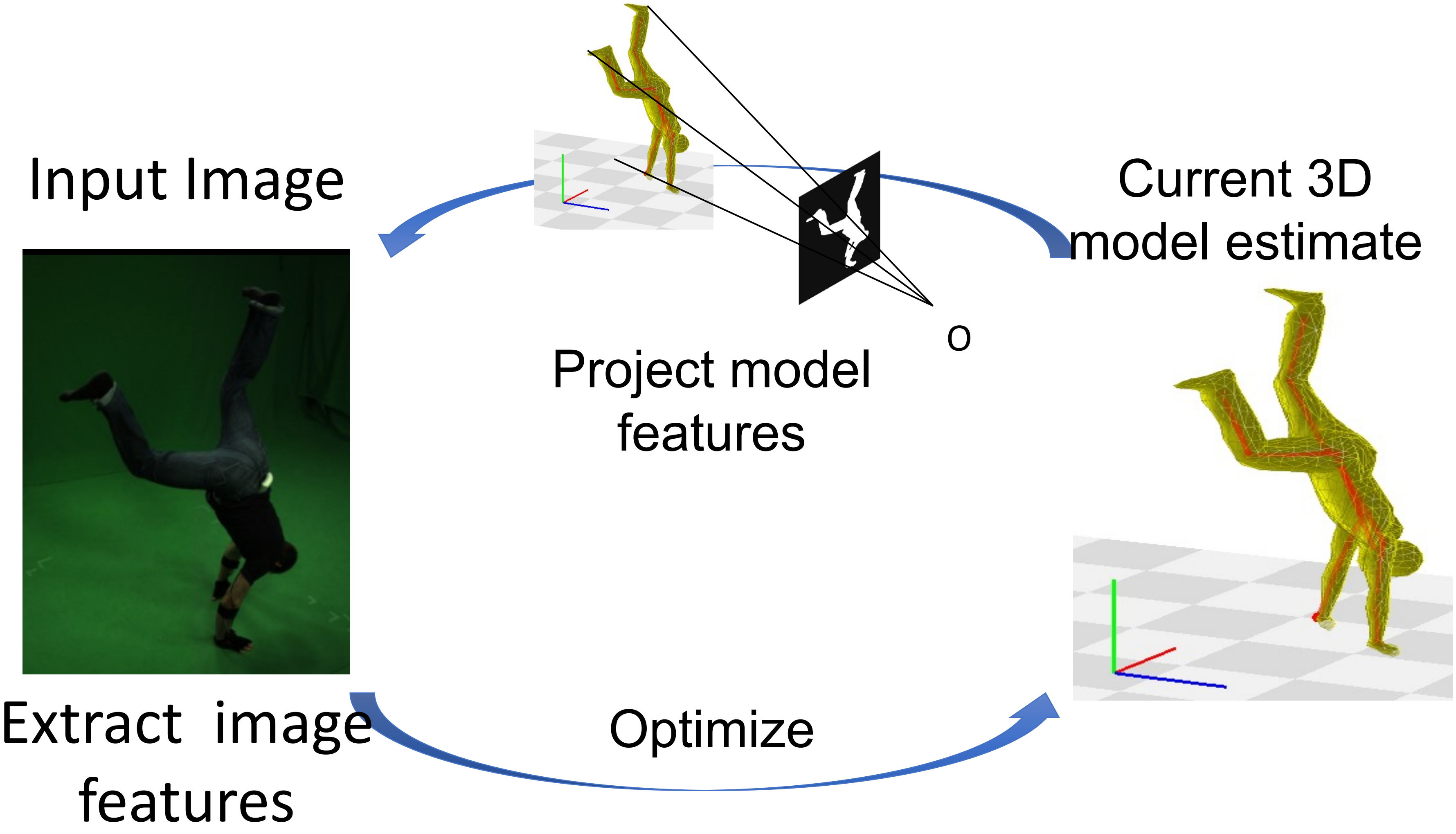
What are good features for fitting?



How to "match" 3D SMPL with 2D image?

What objective should we optimize?

How to "match" a 3D Model with a 2D image?



Pons-Moll et al. CVPR'10
Pons-Moll et al. 2011 Model Based Pose Estimation

How to "match" 3D SMPL with 2D image?

Check your understanding:

- Should we match model to image or image to model?
  - What if the images contain partial body?

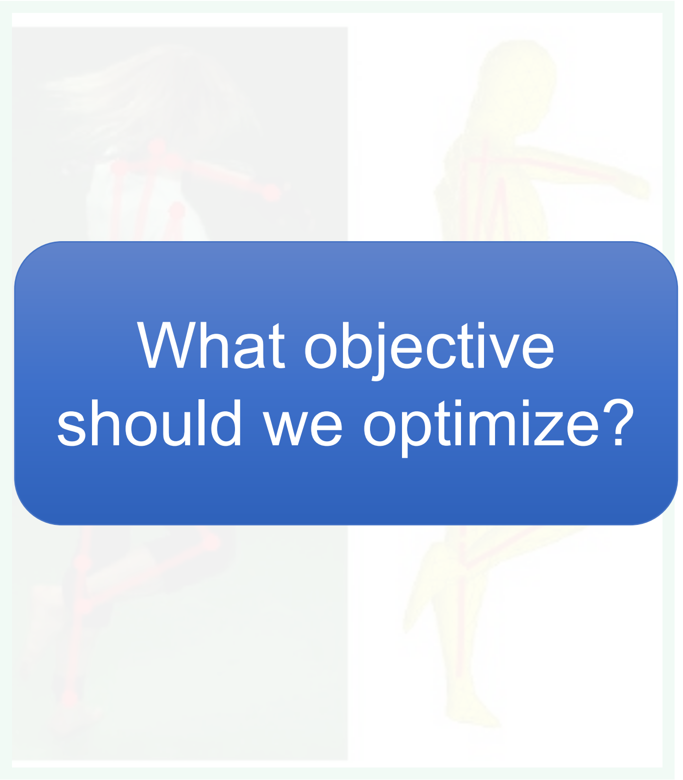
- What if our features contain outliers?
  We saw in our assignments that optimization can be susceptible to outliers.

General framework for optimization

1. Extract features
2. Predict and match
3. Optimize

What are good features for fitting?

How to "match" 3D SMPL with 2D image?



What objective should we optimize?

What objective should we optimize?

A lot of problems can be formulated as **Non-Linear Least Squares**

$$e(\mathbf{x}_t) = \sum_{i=1}^{N} e_i(\mathbf{x}_t) = \sum_{i=1}^{N} \|\tilde{\mathbf{r}}_i(\mathbf{x}_t) - \mathbf{r}_i\|^2$$

SMPL params. for image *t*

Sum over all *N* features

2D image features predicted by the model

Observed 2D features

What objective should we optimize?

A lot of problems can be formulated as **Non-Linear Least Squares**

$$e(\mathbf{x}_t) = \sum_{i=1}^{N} e_i(\mathbf{x}_t) = \sum_{i=1}^{N} \|\tilde{\mathbf{r}}_i(\mathbf{x}_t) - \mathbf{r}_i\|^2$$

Assuming errors re independent and Gaussian distributed, least squares is equivalent to a **MAP** estimate

$$p(\mathbf{x}_t | \mathbf{y}_t) \propto p(\mathbf{y}_t | \mathbf{x}_t) p(\mathbf{x}_t) \propto \exp\left(-\sum_{i}^{N} \mathbf{e}_i^2(\mathbf{y}_t^i | \mathbf{x}_t) \right) p(\mathbf{x}_t)$$

Observation (image)

Optimization with Least Squares

Express the problem in vector form

$$e(\mathbf{x}_t) = \mathbf{e}^T\mathbf{e} \qquad \mathbf{e} \in \mathbb{R}^{2N}$$

$$\mathbf{e} = (\mathbf{e}_1^T, \mathbf{e}_2^T, \ldots, \mathbf{e}_N^T)$$

$$e(\mathbf{x}_t) = \begin{bmatrix} \underbrace{\Delta r_{1,x} \quad \Delta r_{1,y}}_{\text{Residual for match 1}} & \ldots & \Delta r_{N,x} & \Delta r_{N,y} \end{bmatrix} \begin{bmatrix} \Delta r_{1,x} \\ \Delta r_{1,y} \\ \vdots \\ \vdots \\ \Delta r_{N,x} \\ \Delta r_{N,y} \end{bmatrix}$$

Optimization with Least Squares

$$\begin{aligned}\Delta\mathbf{x} &= \arg\min_{\Delta\mathbf{x}} \frac{1}{2}\, \mathbf{e}^T(\mathbf{x}_t + \Delta\mathbf{x})\ \mathbf{e}(\mathbf{x}_t + \Delta\mathbf{x}) \\ &= \arg\min_{\Delta\mathbf{x}} \frac{1}{2}(\mathbf{e} + \mathbf{J}_t\Delta\mathbf{x})^T(\mathbf{e} + \mathbf{J}_t\Delta\mathbf{x}) \\ &= \arg\min_{\Delta\mathbf{x}} \frac{1}{2}\mathbf{e}^T\mathbf{e} + \Delta\mathbf{x}^T\underbrace{\mathbf{J}_t^T\mathbf{e}}_{\text{Gradient}} + \frac{1}{2}\Delta\mathbf{x}^T\underbrace{\mathbf{J}_t^T\mathbf{J}_t}_{\sim\text{Hessian}}\Delta\mathbf{x}\end{aligned}$$

$$\boxed{\Delta\mathbf{x} = -\left(\mathbf{J}_t^T\mathbf{J}_t + \mu\mathbf{I}\right)^{-1}\mathbf{J}_t^T\mathbf{e}}$$

Take a step in that direction $\qquad \mathbf{x}_{t+1} = \mathbf{x}_t + \Delta\mathbf{x}$

The land of optimization if full of pitfalls

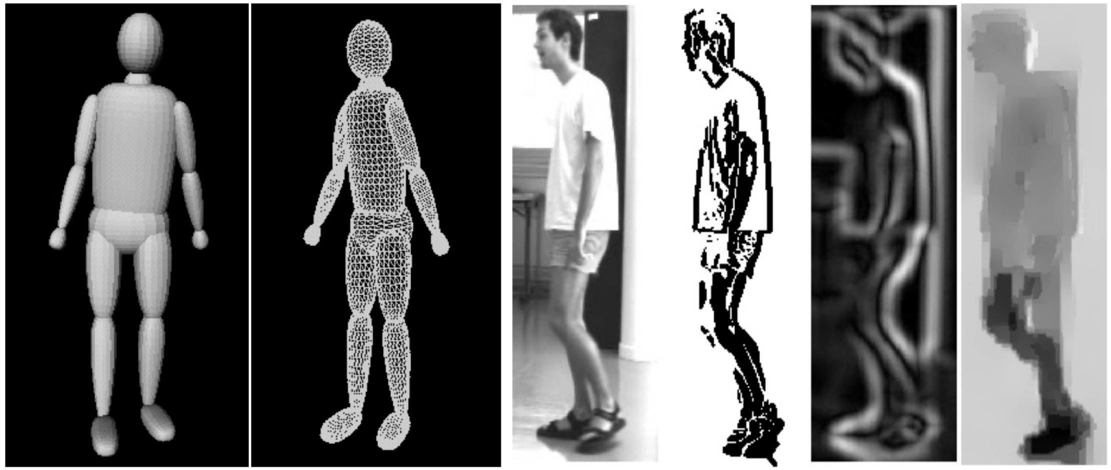
$$\Delta \mathbf{x} = -(\mathbf{J}_t^T \mathbf{J}_t + \mu \mathbf{I})^{-1} \mathbf{J}_t^T \mathbf{e}$$

~Hessian
(approximation)

Gradient term
of total error

Jacobian of the vector error
(entries often correspond
to vertices in the model)

Example: Covariance scaled sampling for Monocular 3D body tracking



$$p(\mathbf{x}|\bar{\mathbf{r}}) \;\propto\; p(\bar{\mathbf{r}}|\mathbf{x})\, p(\mathbf{x}) \;=\; \exp\left(-\textstyle\int e(\bar{\mathbf{r}}_i|\mathbf{x})\, di\right)\; p(\mathbf{x})$$

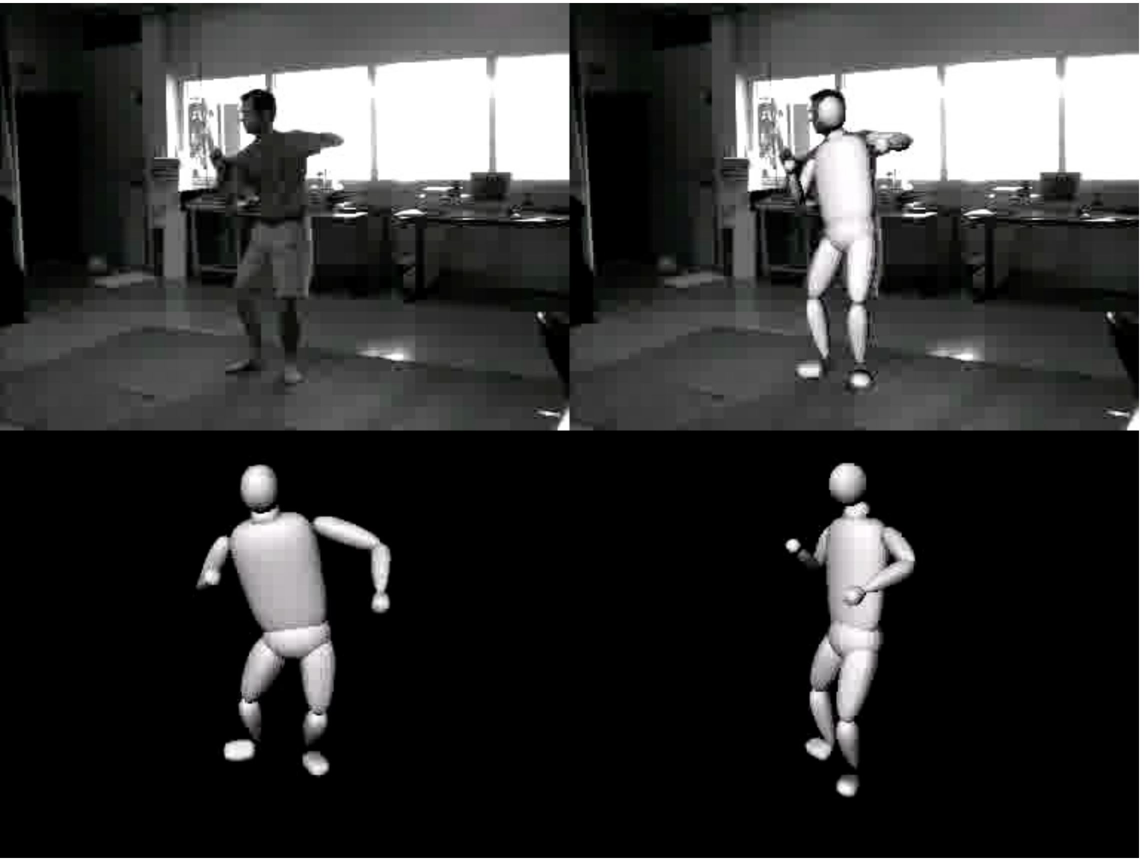
$$e_i(\mathbf{x}) = \begin{cases} \frac{1}{2}\rho_i(\mathbf{\Delta r}_i(\mathbf{x})\, \mathbf{W}_i\, \mathbf{\Delta r}_i(\mathbf{x})^\top) & \text{if } i \text{ is assigned} \\ \nu_{bf} = \nu & \text{if back-facing} \\ \nu_{occ} = k\nu, \;\; k > 1 & \text{if occluded} \end{cases}$$

Features:

- Motion boundaries
- Edges
- Optical Flow

Sminchisescu et al. CVPR’01

Example: Covariance scaled sampling for Monocular 3D body tracking



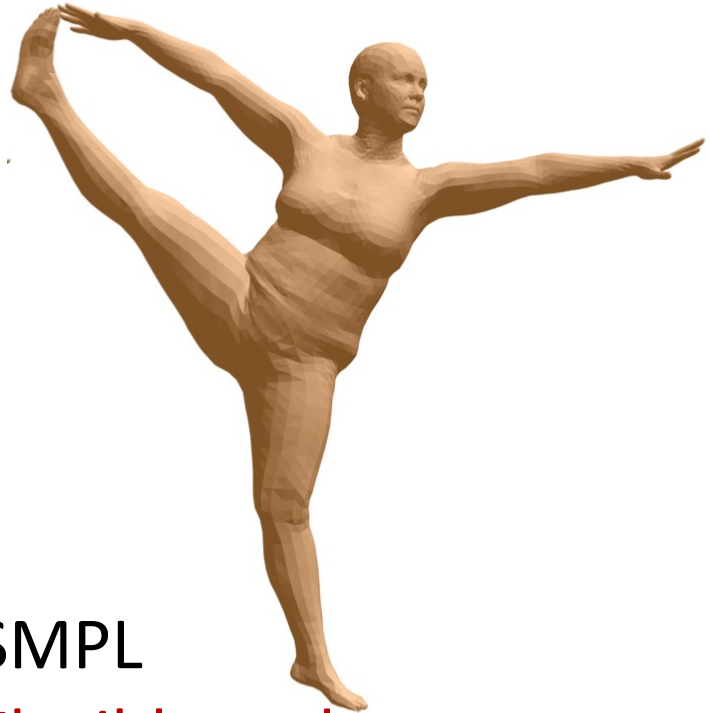
- Remarkable for **2001**!!

- Constrained to lab settings
- Not robust to difficult poses and backgrounds

So what has changed in almost 20 years

2D pose detection works very reliably!

A very good feature!! → why?



SMPL

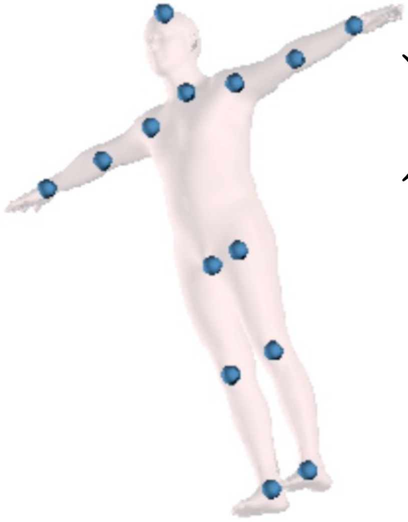
Flexible and easy to use model which adapts to different shapes

How does SMPLify execute the general framework to fit SMPL to an image

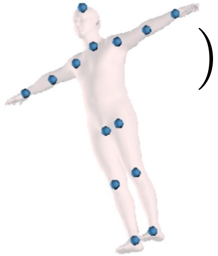
1. Extract features
2. Predict and match
3. Optimize

2D joints are quite reliable

Match projection of 3D joints with 2D joints

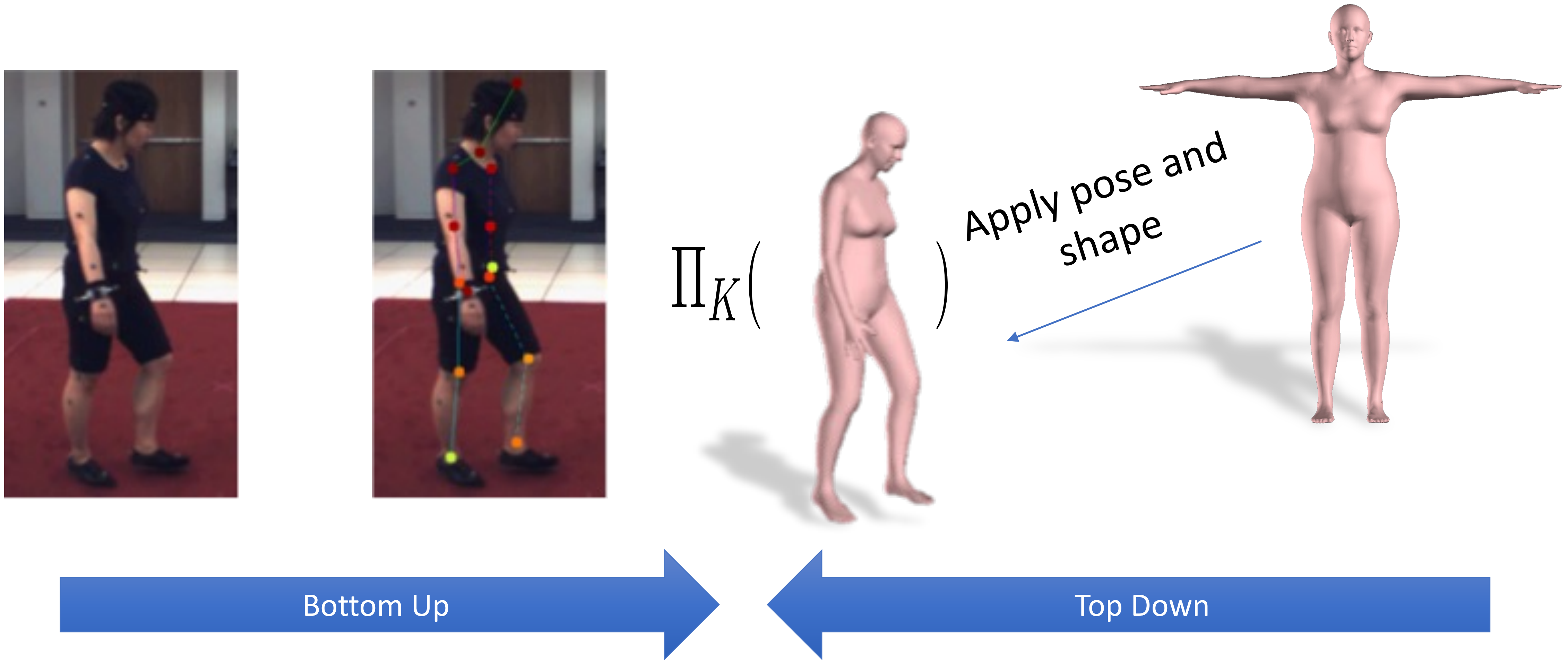
$\Pi_K($  $)$

Min. L_2 dist. b/w image and projected joints

$\|$  $- \Pi_K($  $) \|_2^2$

Bogo et al., ECCV'16

Bottom up and top down should match



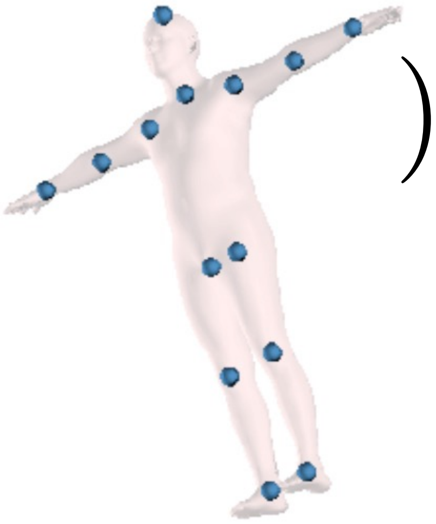
Bottom up and top down should match

shape pose camera 2D joints [1]

$$E_J(\beta, \theta, K; J_{est}) = \Big\| \quad\quad - \Pi_K(\quad\quad) \Big\|_2^2$$

Bottom-up
2D joints[1]

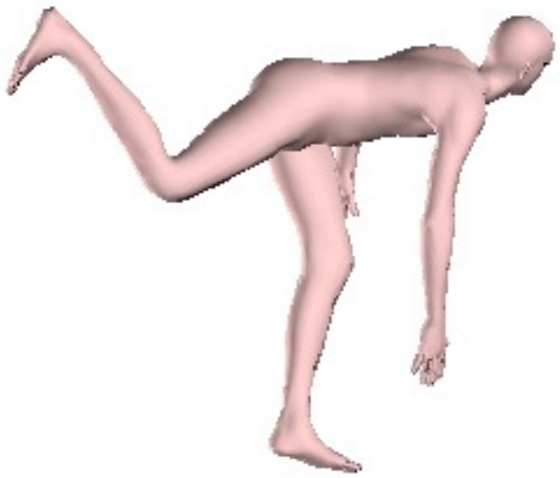
Camera Projection



Top-down
SMPL fit

[1] - Pishchulin et al., "DeepCut: Joint subset partition and labeling for multi person pose estimation", CVPR 2016

SMPLify Objective Function



camera joints

$$E(\vec{\beta}, \vec{\theta}, K; J_{est}) = \quad E_J(\vec{\beta}, \vec{\theta}, K; J_{est})$$

Data term

Is this enough?

Problem: Depth Ambiguity

Input image

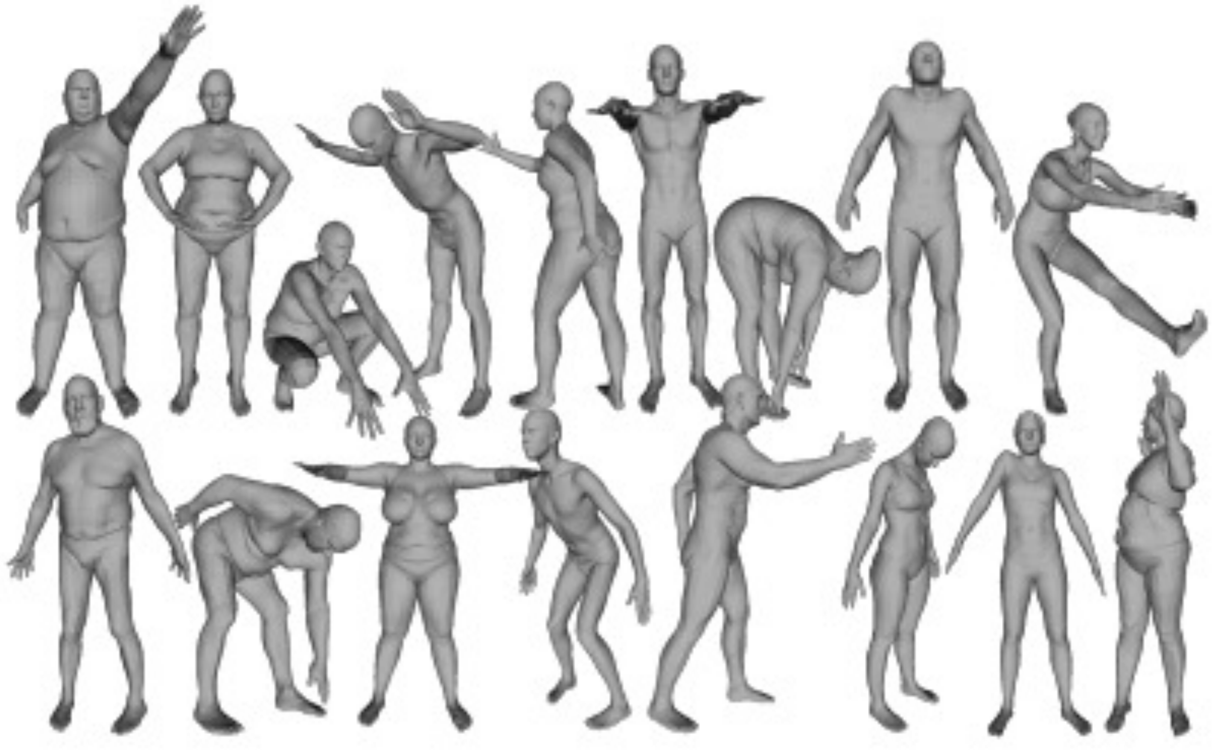
SMPL aligns well with the image

Side view

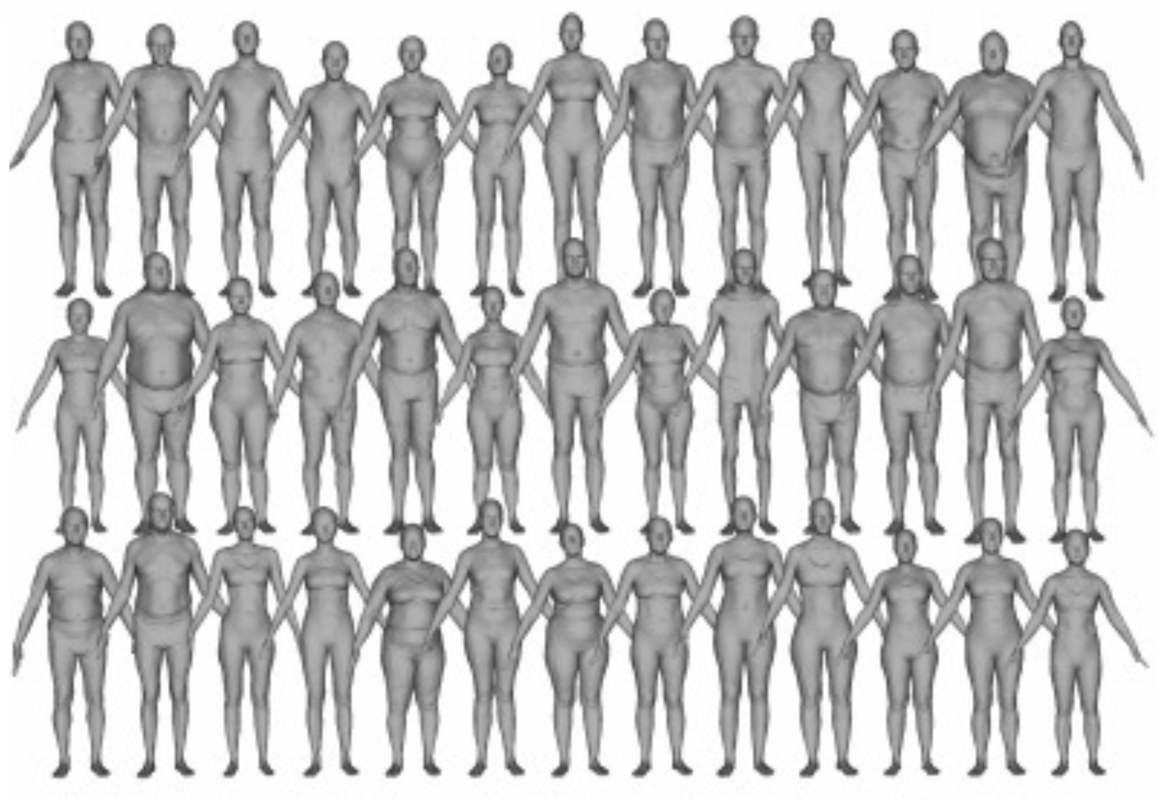
Side view is incorrect

[Guan et al., Estimating human shape and pose from a single image. ICCV 2009.]

Solution: Pose and Shape Priors

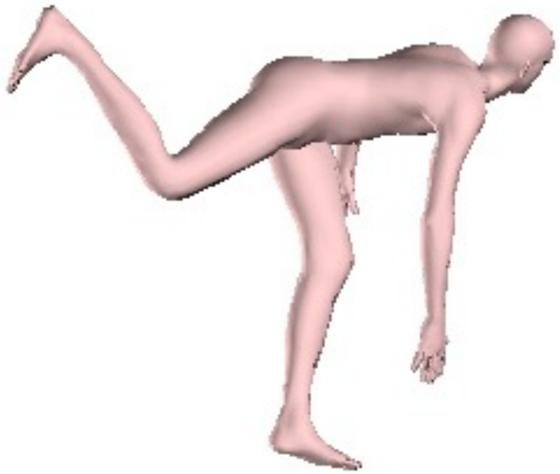


$E_\theta(\vec{\theta})$ Pose Prior



$E_\beta(\vec{\beta})$ Shape Prior

Updated SMPLify Objective Function



camera joints

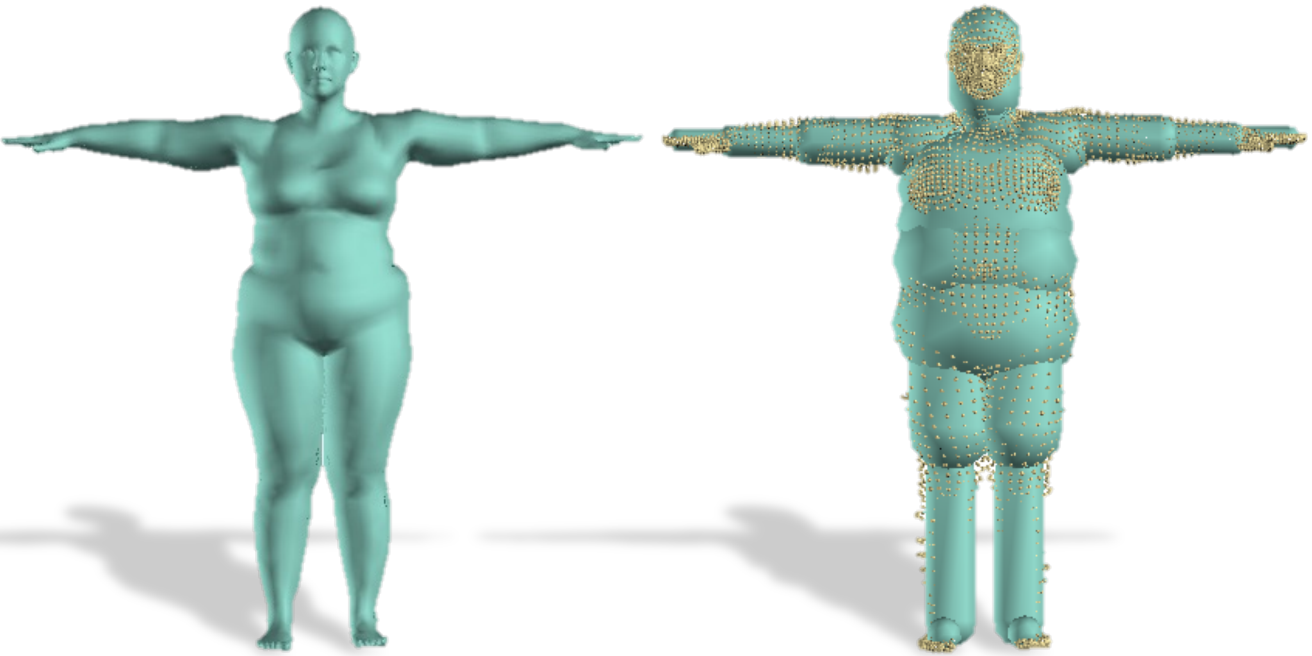
$$E(\vec{\beta}, \vec{\theta}, K; J_{est}) =$$

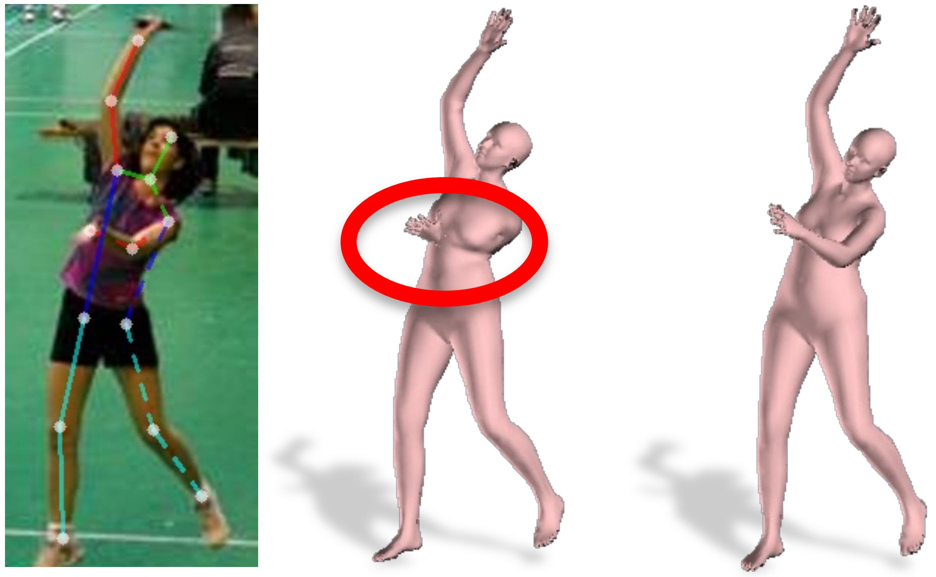
$$E_J(\vec{\beta}, \vec{\theta}, K; J_{est}) + E_a(\vec{\theta}) + E_\theta(\vec{\theta}) + E_\beta(\vec{\beta})$$

Data term | Prior on unnatural joint bending | Prior on pose | Prior on shape

Problem: Interpenetrations

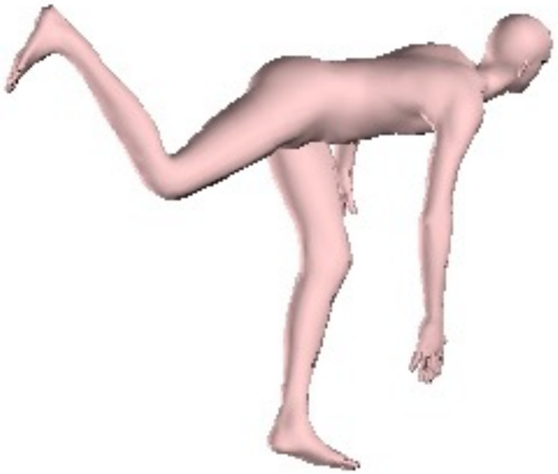
Solution: Approx. surface with capsules and penalise intersections





Before After

SMPLify Objective Function



pose and shape priors

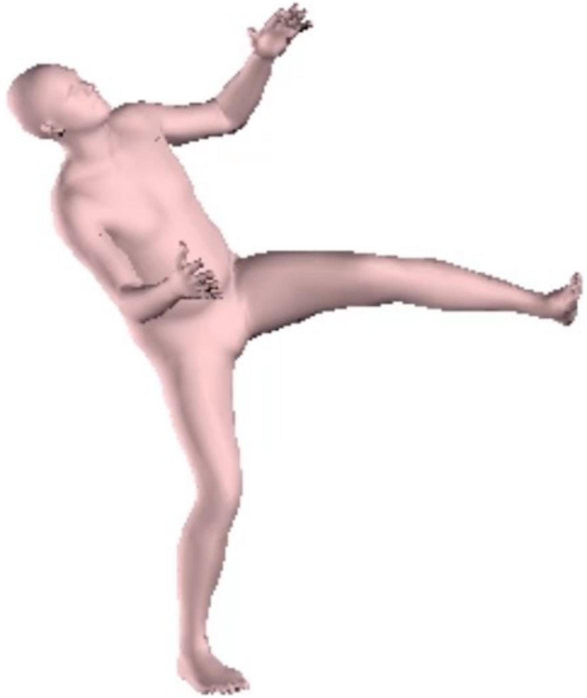
$$E(\vec{\beta}, \vec{\theta}, K; J_{est}) =$$

$$E_J(\vec{\beta}, \vec{\theta}, K; J_{est}) + E_a(\vec{\theta}) + E_\theta(\vec{\theta}) + E_{sp}(\vec{\theta}, \vec{\beta}) + E_\beta(\vec{\beta})$$

Joint projection error

interpenetration

Results on Leeds Sports Poses (LSP)



Datasets

Indoor, lab

Outdoor, unconstrained

Human-Eva H3.6M

3DPW

CoCo, Leeds,etc

With reference pose and shape

Into the wild

SMPLify-X vs SMPLify

- SMPLify-X improves on SMPLify.

- Key changes:
  - Upgrade SMPL to have detailed hands and face (SMPL-X).
  - Update the pose priors from GMM to VAE.
  - Train classifier to predict gender and select model accordingly.

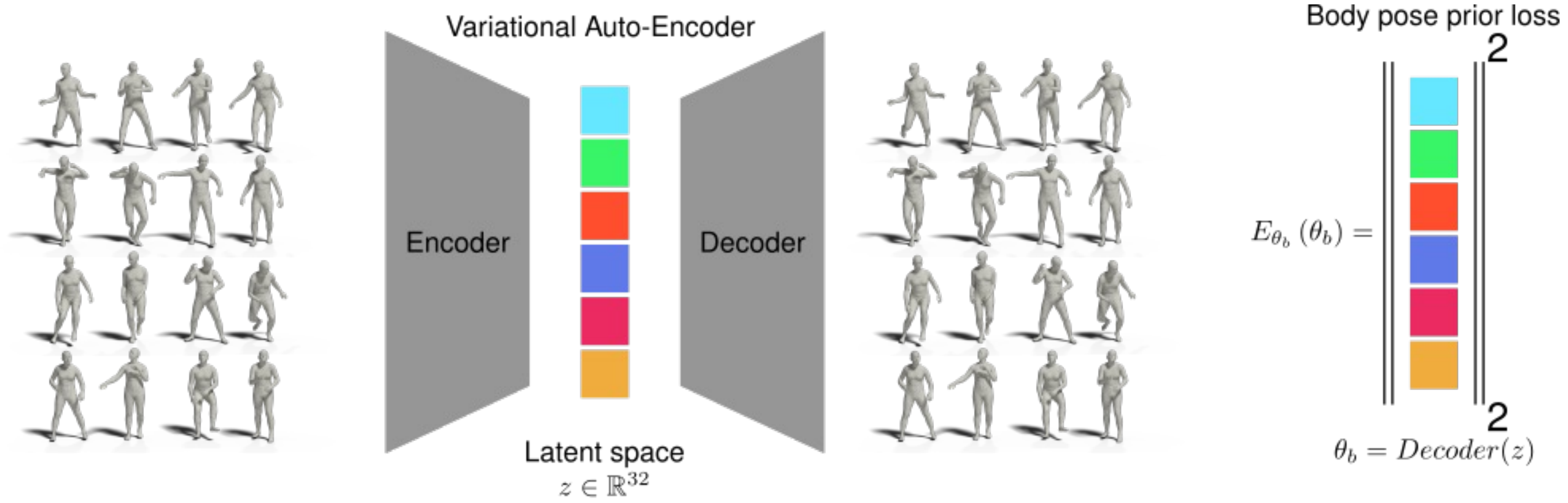
- The key ideas remain the same.

SMPLify-X. Pavlakos et al. CVPR'19

Add detailed hands and face to SMPL

SMPLify-X. Pavlakos et al. CVPR'19

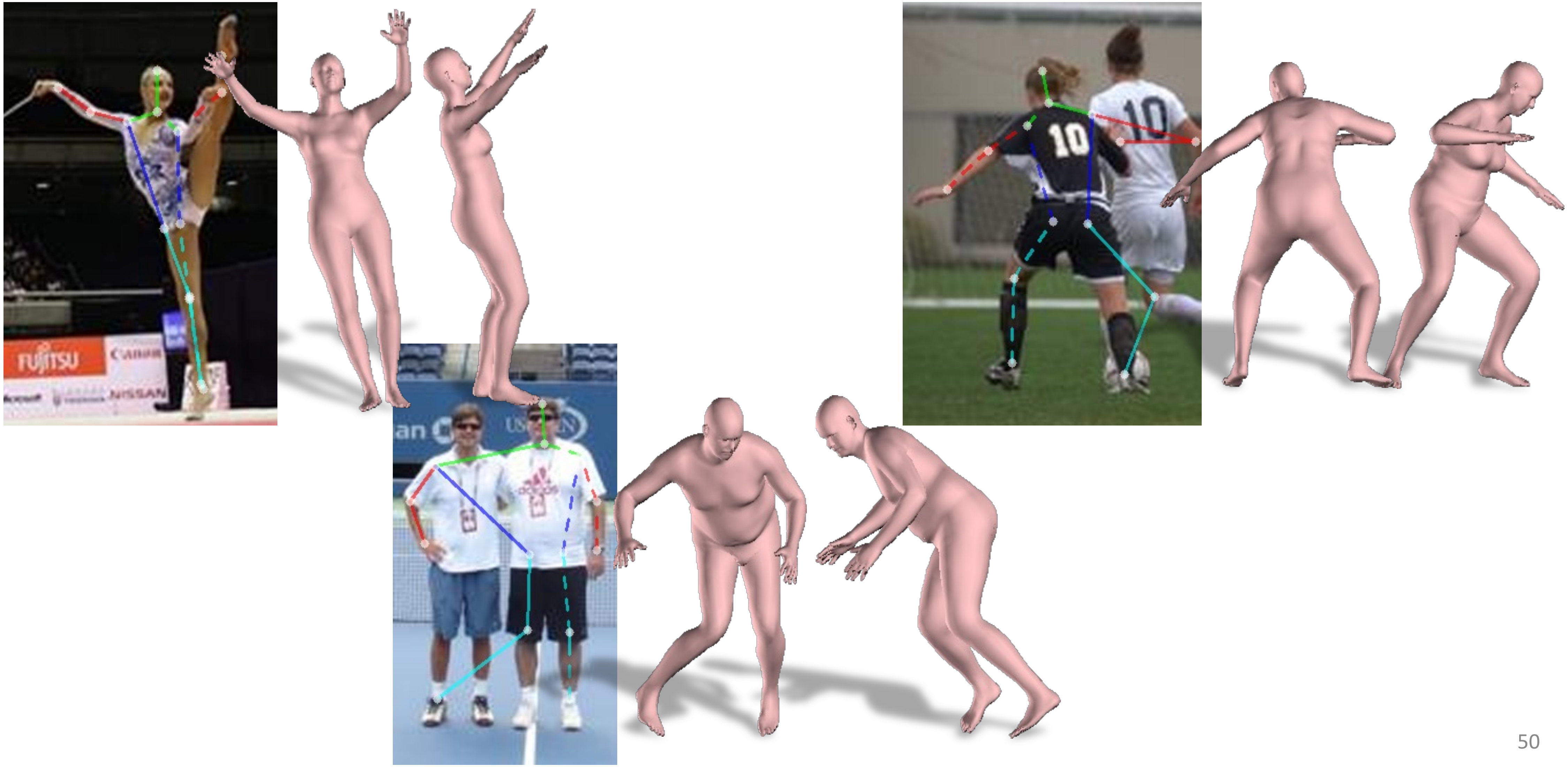
Use a VAE to learn the manifold of poses



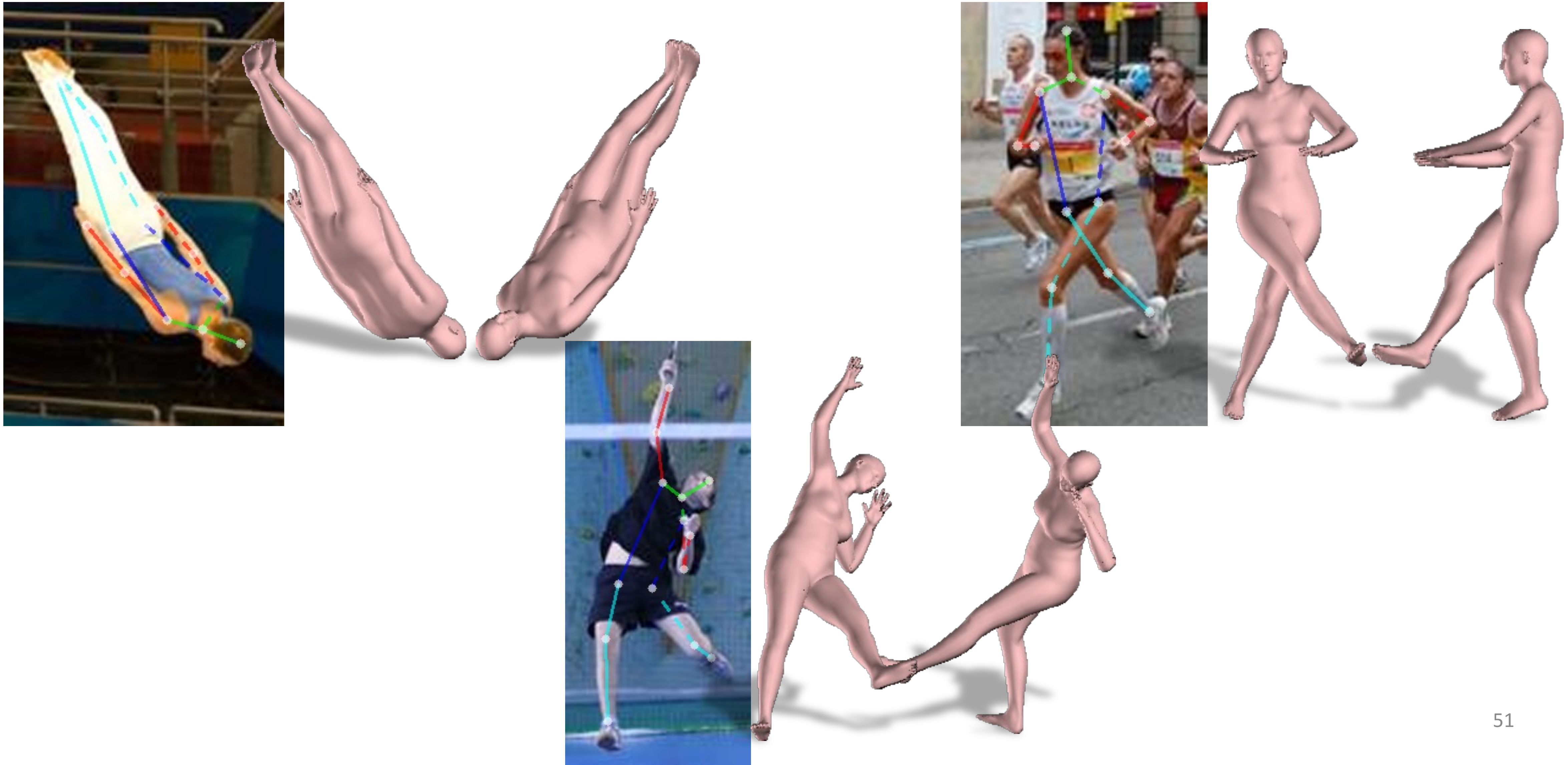
SMPLify-X captures hands and faces better

Limitations of Optimization

Failure modes: 2D CNN failure



Failure modes: Depth Ambiguity



Pros/ Cons of local optimization

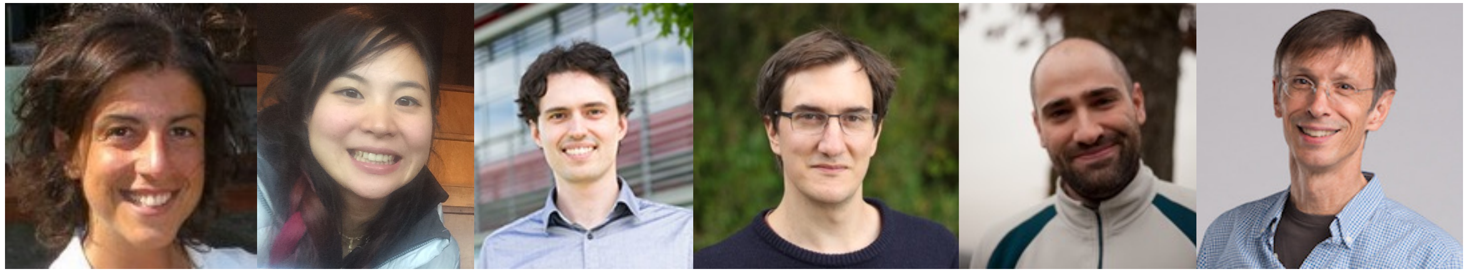
- ✓ It (can be) fast and accurate
- ✗ Prone to local minima
- ✗ Requires initialization
- ✗ Matching cost is ambiguous

Can we use learning to solve these?

Thanks!

- Can we use learning to address some of these limitations?

- More in Part 2…



Keep it SMPL: Automatic Estimation of 3D Human Pose and Shape from a Single Image (SMPLify)

F. Bogo*, A. Kanazawa*, C. Lassner, P. Gehler, J. Romero, M. J. Black

ECCV'16