Universidad Rey Juan Carlos

MSLab
Multimodal Simulation Lab

Understanding Parametric Human Body Models

SCA 2026 - Graduate Summer School

Marc Comino (marc.comino@urjc.es)

http://www.mslab.es/

Motivation

Where do we come from?

Where are we now?

Why does this matter?

Where do we come from?

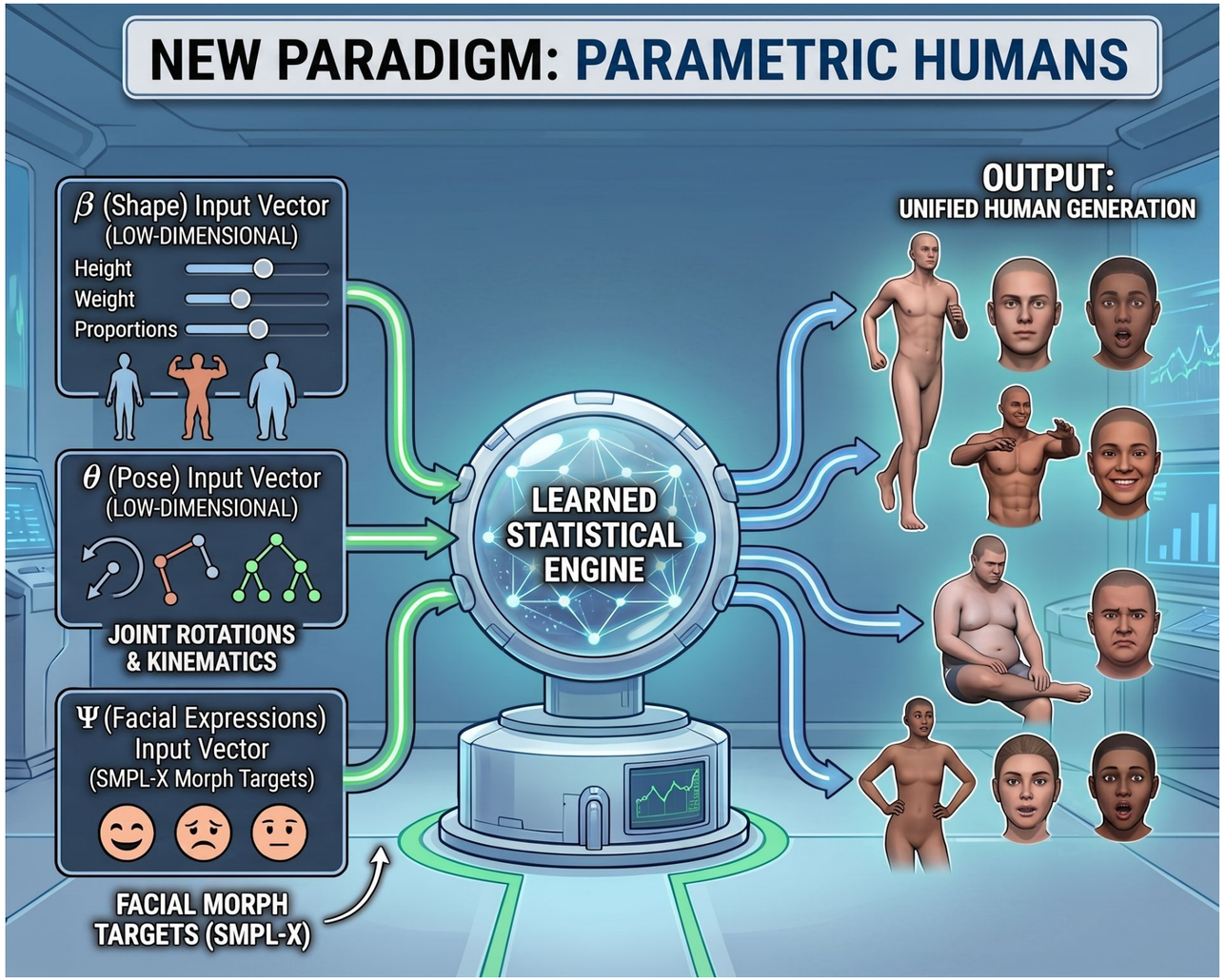
Marc Comino
marc.comino@urjc.es

Where are we now?

Results Obtained using: Velasquez, H. C., Yiannakidis, A., Shin, S., Becherini, G., Höschle, M., Tesch, J., ... & Black, M. J. (2026). MAMMA: Markerless Accurate Multi-person Motion Acquisition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 7175-7186).

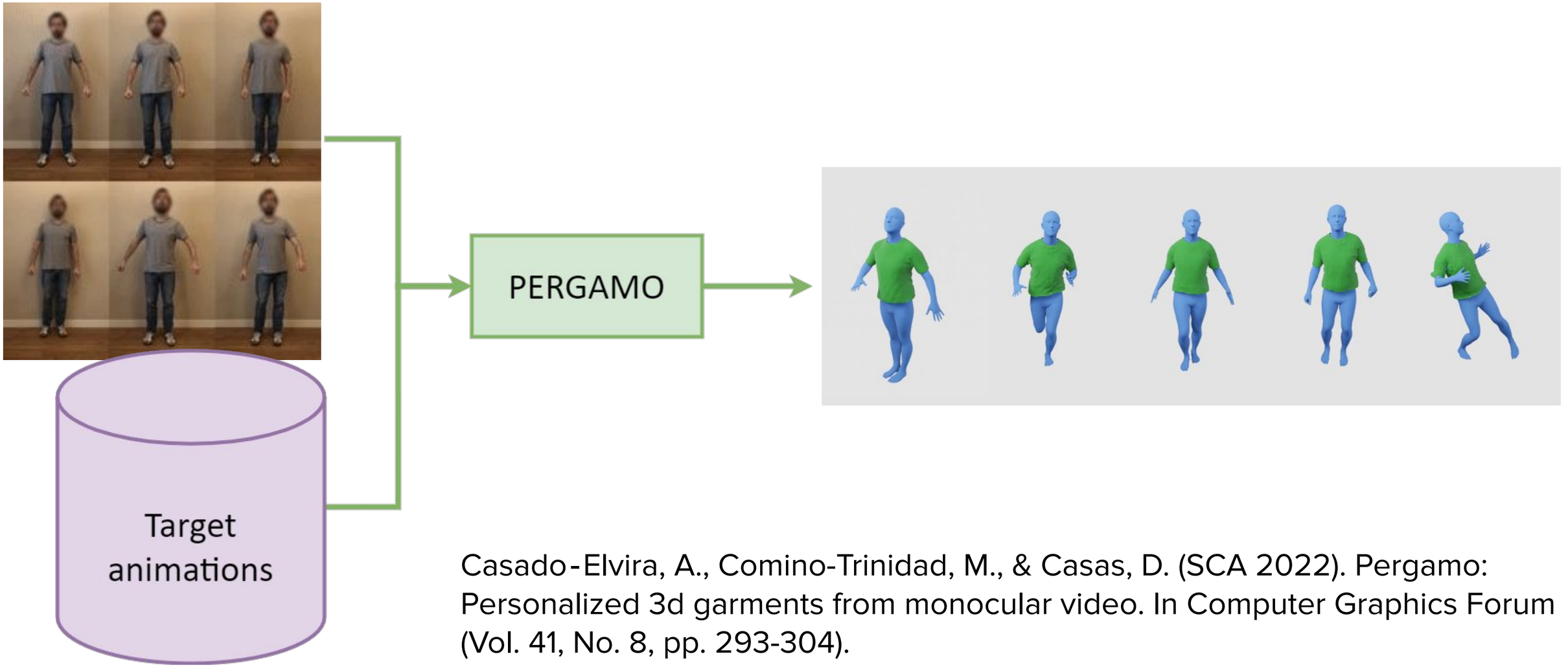
Marc Comino
marc.comino@urjc.es

Where are we now?



Marc Comino
marc.comino@urjc.es

Why does this matter?



Casado-Elvira, A., Comino-Trinidad, M., & Casas, D. (SCA 2022). Pergamo: Personalized 3d garments from monocular video. In Computer Graphics Forum (Vol. 41, No. 8, pp. 293-304).

Marc Comino
marc.comino@urjc.es

Why does this matter?

Marc Comino
marc.comino@urjc.es

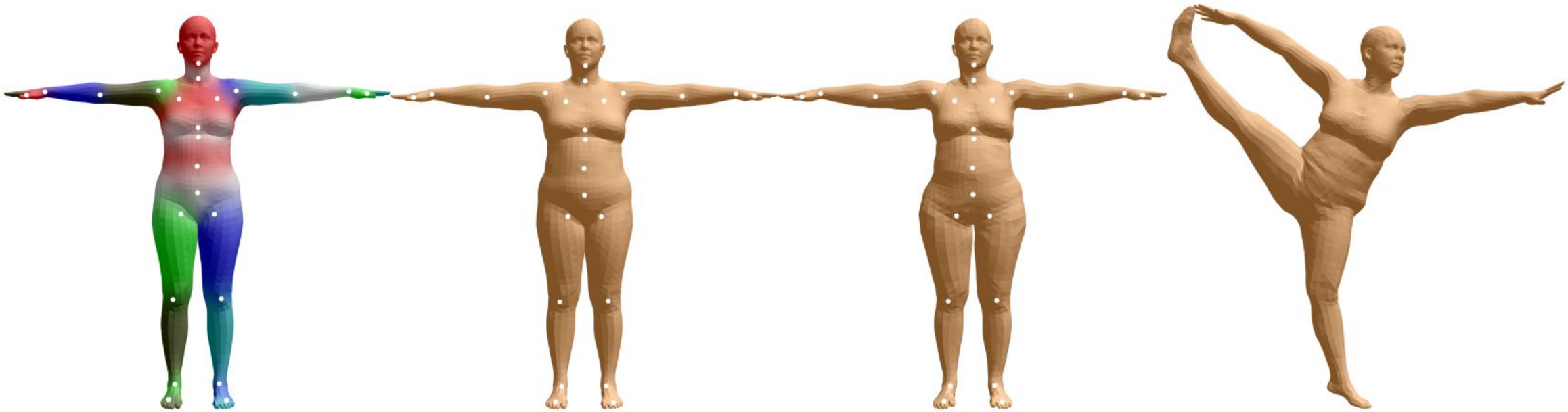
Parametric Human Models

The SMPL Model

The SMPL Model: Step by Step

Shifting to SMPLX

The SMPL Model



Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., & Black, M. J. (2015). SMPL: a skinned multi-person linear model. ACM Transactions on Graphics (TOG), 34(6), 1-16.

Marc Comino
marc.comino@urjc.es

The SMPL Model: Step by Step

T (Template Mesh):

The zero-pose, neutral rest mesh representing the mean human body shape, consisting of 6,890 vertices for SMPL and expanded to 10,475 vertices for SMPL-X.

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

Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., & Black, M. J. (2015). SMPL: a skinned multi-person linear model. ACM Transactions on Graphics (TOG), 34(6), 1-16.

Marc Comino
marc.comino@urjc.es

The SMPL Model: Step by Step

T (Template Mesh):

The zero-pose, neutral rest mesh representing the mean human body shape, consisting of 6,890 vertices for SMPL and expanded to 10,475 vertices for SMPL-X.

```python
import bpy
import numpy as np

# Path to your extracted SMPL-X model file (e.g., SMPLX_NEUTRAL.npz)
model_path = "/home/marcct/Documents/SMPL/SMPLX/SMPLX_NEUTRAL_2020.npz"
data = np.load(model_path, allow_pickle=True)

# Variable names matching mathematical SMPL notation
T_bar = data['v_template']          # \bar{T} : Base rest template vertices
F = data['f']                       # F : Face topology indices
W = data['weights']                 # \mathcal{W} : Linear blend skinning weights matrix
kintree = data['kintree_table'][0]  # Kinematic tree parent indices for hierarchy
```

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

Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., & Black, M. J. (2015). SMPL: a skinned multi-person linear model. ACM Transactions on Graphics (TOG), 34(6), 1-16.

Marc Comino
marc.comino@urjc.es

The SMPL Model: Step by Step

W (Skinning Blend Weights):

A fixed matrix computed offline by minimizing the error between a deformed template and thousands of multi-pose 3D scans (from datasets like MPI D-FAUST) to find the optimal surface influence for each joint.

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

Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., & Black, M. J. (2015). SMPL: a skinned multi-person linear model. ACM Transactions on Graphics (TOG), 34(6), 1-16.

Marc Comino
marc.comino@urjc.es

The SMPL Model: Step by Step

W (Skinning Blend Weights):

A fixed matrix computed offline by minimizing the error between a deformed template and thousands of multi-pose 3D scans (from datasets like MPI D-FAUST) to find the optimal surface influence for each joint.

```python
import bpy
import numpy as np

# Path to your extracted SMPL-X model file (e.g., SMPLX_NEUTRAL.npz)
model_path = "/home/marcct/Documents/SMPL/SMPLX/SMPLX_NEUTRAL_2020.npz"
data = np.load(model_path, allow_pickle=True)

# Variable names matching mathematical SMPL notation
T_bar = data['v_template']          # \bar{T} : Base rest template vertices
F = data['f']                       # F : Face topology indices
W = data['weights']                 # \mathcal{W} : Linear blend skinning weights matrix
kintree = data['kintree_table'][0]  # Kinematic tree parent indices for hierarchy
```

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

Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., & Black, M. J. (2015). SMPL: a skinned multi-person linear model. ACM Transactions on Graphics (TOG), 34(6), 1-16.

Marc Comino
marc.comino@urjc.es

The SMPL Model: Step by Step

Bs (Shape Blendshapes):

A fixed displacement matrix calculated via Principal Component Analysis (PCA) on thousands of aligned 3D body scans from the CAESAR dataset to capture major structural identity variations like height and weight.

10 Betas (The Research Default)

300 Betas (The Full Spectrum)

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

Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., & Black, M. J. (2015). SMPL: a skinned multi-person linear model. ACM Transactions on Graphics (TOG), 34(6), 1-16.

Marc Comino
marc.comino@urjc.es

The SMPL Model: Step by Step

Bs (Shape Blendshapes):

A fixed displacement matrix calculated via Principal Component Analysis (PCA) on thousands of aligned 3D body scans from the CAESAR dataset to capture major structural identity variations like height and weight.

10 Betas (The Research Default)

300 Betas (The Full Spectrum)

```
# Variable names matching mathematical SMPL notation
T_bar = data['v_template']          # \bar{T} : Base rest template vertices
W = data['weights']                 # \mathcal{W} : Linear blend skinning weights matrix
kintree = data['kintree_table'][0]  # Kinematic tree parent indices for hierarchy
F = data['f']                       # F : Face topology indices

# Change these values to test different body shapes/identities
Betas = np.array([1.5, -0.8, 2.0, 0.5, -1.2, 0.3, -0.5, 1.1, -0.2, 0.7])

# Extract the shape blendshapes matrix B_S
B_s = data['shapedirs'][:, :, :10]
```

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

Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., & Black, M. J. (2015). SMPL: a skinned multi-person linear model. ACM Transactions on Graphics (TOG), 34(6), 1-16.

Marc Comino
marc.comino@urjc.es

The SMPL Model: Step by Step

J(β) (Joints of Beta):

Represents the rest-pose world coordinates of the body's skeletal joints, mathematically calculated by multiplying a learned joint regressor matrix against the shape-deformed rest template mesh.

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

Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., & Black, M. J. (2015). SMPL: a skinned multi-person linear model. ACM Transactions on Graphics (TOG), 34(6), 1-16.

Marc Comino
marc.comino@urjc.es

The SMPL Model: Step by Step

J(β) (Joints of Beta):

Represents the rest-pose world coordinates of the body's skeletal joints, mathematically calculated by multiplying a learned joint regressor matrix against the shape-deformed rest template mesh.

```python
# Extract the shape blendshapes matrix B_S
B_s = data['shapedirs'][:, :, :10]

# Decode \beta into Template mesh displacements via dot product
v_displacements = np.dot(B_s, Betas)
T_shaped = T_bar + v_displacements

# Compute joint locations J from the base template using the regressor matrix \mathcal{J}
J_regressor = data['J_regressor']
if hasattr(J_regressor, 'toarray'):
    J_regressor = J_regressor.toarray()
J = np.dot(J_regressor, T_shaped)      # J(\beta) in rest pose where shape coefficients \be
```

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

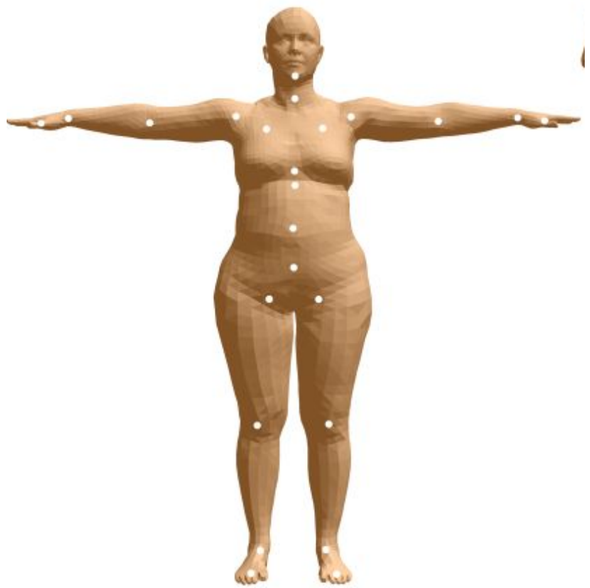
Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., & Black, M. J. (2015). SMPL: a skinned multi-person linear model. ACM Transactions on Graphics (TOG), 34(6), 1-16.

Marc Comino
marc.comino@urjc.es

The SMPL Model: Step by Step

Bp (Pose Blendshapes):

An optimized matrix of 3D vertex displacements multiplied linearly by the 207 elements of the joints' relative 3 x 3 rotation matrices (9 x 23 joints) to counteract linear skinning artifacts and preserve muscle volume during bending.



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

Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., & Black, M. J. (2015). SMPL: a skinned multi-person linear model. ACM Transactions on Graphics (TOG), 34(6), 1-16.

Marc Comino
marc.comino@urjc.es

The SMPL Model: Step by Step

Bp (Pose Blendshapes):

An optimized matrix of 3D vertex displacements multiplied linearly by the 207 elements of the joints' relative 3 x 3 rotation matrices (9 x 23 joints) to counteract linear skinning artifacts and preserve muscle volume during bending.

```
# Extract the pose blendshapes matrix B_P to calculate volume corrections
B_p = data['posedirs']

# Dynamically determine the number of pose-driven joints from the tensor shape
NumJoints = B_p.shape[2] // 9

# Initialize the pose vector array for all joints including the root
Thetas = np.zeros((NumJoints + 1, 3))

# Apply a rotation to the right hip which is joint index 2 in the standard hierarchy
Thetas[0] = np.array([0, -np.pi/2, 0])
Thetas[1] = np.array([-np.pi/2, 0.0, 0.0])
```

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

Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., & Black, M. J. (2015). SMPL: a skinned multi-person linear model. ACM Transactions on Graphics (TOG), 34(6), 1-16.

Marc Comino
marc.comino@urjc.es

Shifting to SMPLX

1. Main Body Core (Indices 0 to 21)

These 22 joints are almost identical to the original SMPL model. They control the major skeletal posture.

2. Facial Expressive Joints (Indices 22 to 24)

SMPL-X introduces three joints directly to the head structure to drive facial articulation alongside the expression blendshapes:

3. Left Hand & Fingers (Indices 25 to 39)

The hands are modeled using 15 joints, tracking 3 deformation pivot points for each of the 5 fingers (Thumb, Index, Middle, Ring, Pinky).

4. Right Hand & Fingers (Indices 40 to 54)

This region perfectly mirrors the left hand with another 15 joints:

Pavlakos, G., Choutas, V., Ghorbani, N., Bolkart, T., Osman, A. A., Tzionas, D., & Black, M. J. (2019). Expressive body capture: 3d hands, face, and body from a single image. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 10975-10985).

Marc Comino
marc.comino@urjc.es

Shifting to SMPLX

Basic SMPL

```
output = model(
    betas=torch.zeros(batch_size, 10),
    global_orient=torch.zeros(batch_size, 3),
    body_pose=torch.zeros(batch_size, 63),
    transl=torch.zeros(batch_size, 3)
)
```

SMPLX

```
output = model(
    betas=torch.zeros(batch_size, 10),
    global_orient=torch.zeros(batch_size, 3),
    body_pose=torch.zeros(batch_size, 63),
    left_hand_pose=torch.zeros(batch_size, 45),
    right_hand_pose=torch.zeros(batch_size, 45),
    jaw_pose=torch.zeros(batch_size, 3),
    lefteye_pose=torch.zeros(batch_size, 3),
    righteye_pose=torch.zeros(batch_size, 3),
    expression=torch.zeros(batch_size, 10),
    transl=torch.zeros(batch_size, 3)
)
```

(There could be up to 300 betas and 100 expression parameters)

Pavlakos, G., Choutas, V., Ghorbani, N., Bolkart, T., Osman, A. A., Tzionas, D., & Black, M. J. (2019). Expressive body capture: 3d hands, face, and body from a single image. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 10975-10985).

Marc Comino
marc.comino@urjc.es

Shifting to SMPLX

$B_E(\psi)$ (Expression Blendshapes):

An optimized matrix of 3D vertex displacements multiplied linearly by the 100 elements of the ψ expression parameters to produce rich facial expressions.

Pavlakos, G., Choutas, V., Ghorbani, N., Bolkart, T., Osman, A. A., Tzionas, D., & Black, M. J. (2019). Expressive body capture: 3d hands, face, and body from a single image. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 10975-10985).

Marc Comino
marc.comino@urjc.es

Shifting to SMPLX

$B_E(\psi)$ (Expression Blendshapes):

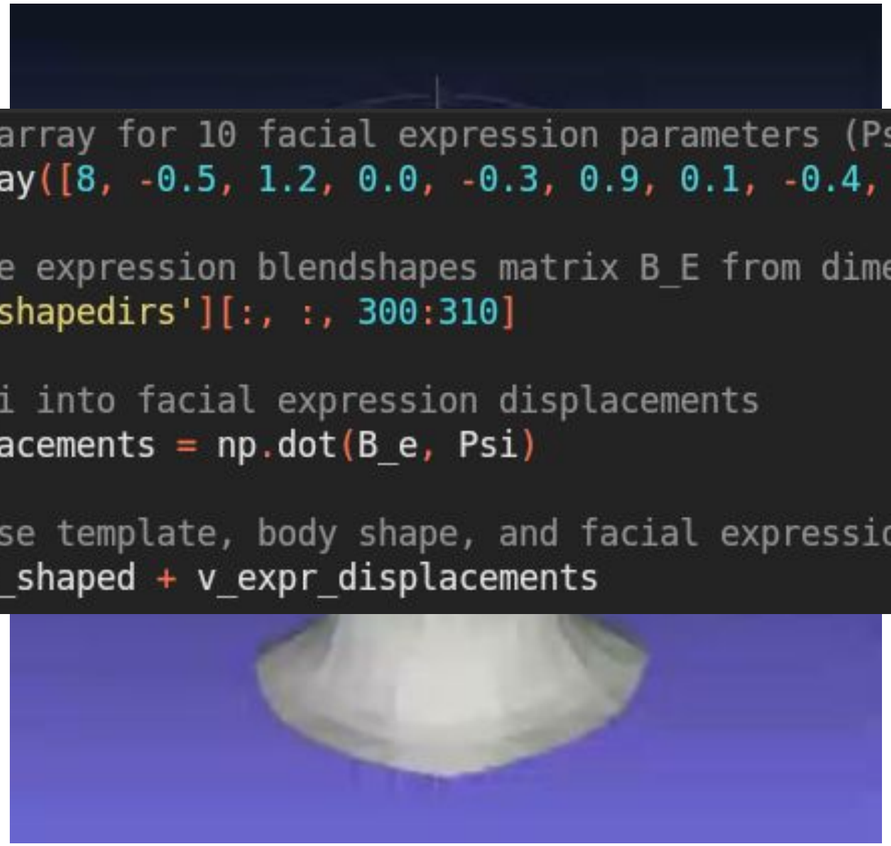
An optimized matrix of 3D vertex displacements multiplied linearly by the 100 elements of the ψ expression parameters to produce rich facial expressions.

```
# Define an array for 10 facial expression parameters (Psi)
Psi = np.array([8, -0.5, 1.2, 0.0, -0.3, 0.9, 0.1, -0.4, 0.5, -0.2])

# Extract the expression blendshapes matrix B_E from dimensions 300 to 310
B_e = data['shapedirs'][:, :, 300:310]

# Decode \psi into facial expression displacements
v_expr_displacements = np.dot(B_e, Psi)

# Combine base template, body shape, and facial expression displacements
T_shaped = T_shaped + v_expr_displacements
```



Pavlakos, G., Choutas, V., Ghorbani, N., Bolkart, T., Osman, A. A., Tzionas, D., & Black, M. J. (2019). Expressive body capture: 3d hands, face, and body from a single image. In Proce edings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 10975-10985).

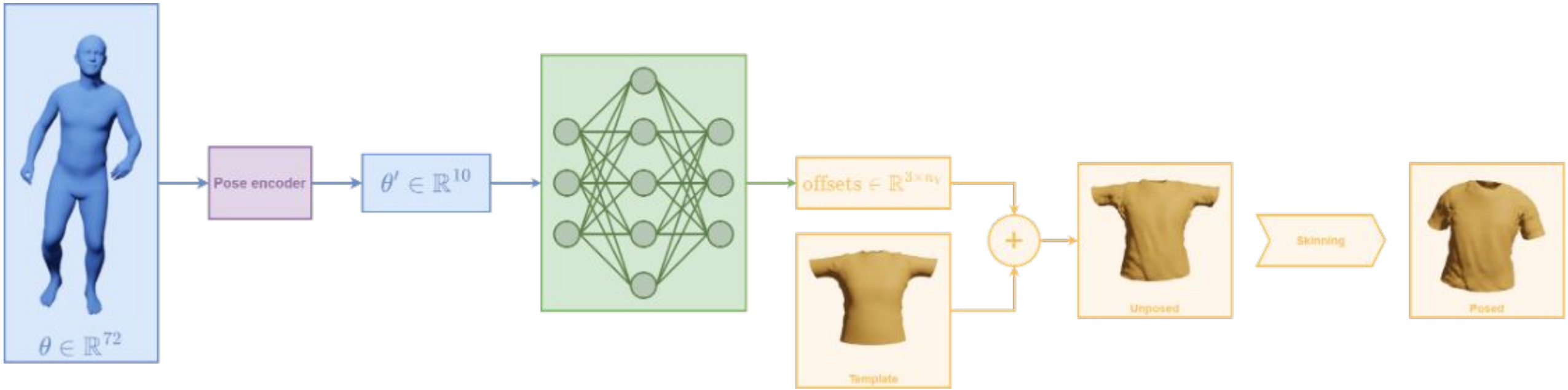
Marc Comino
marc.comino@urjc.es

Other Applications

Dimensionality Reduction

Texturing

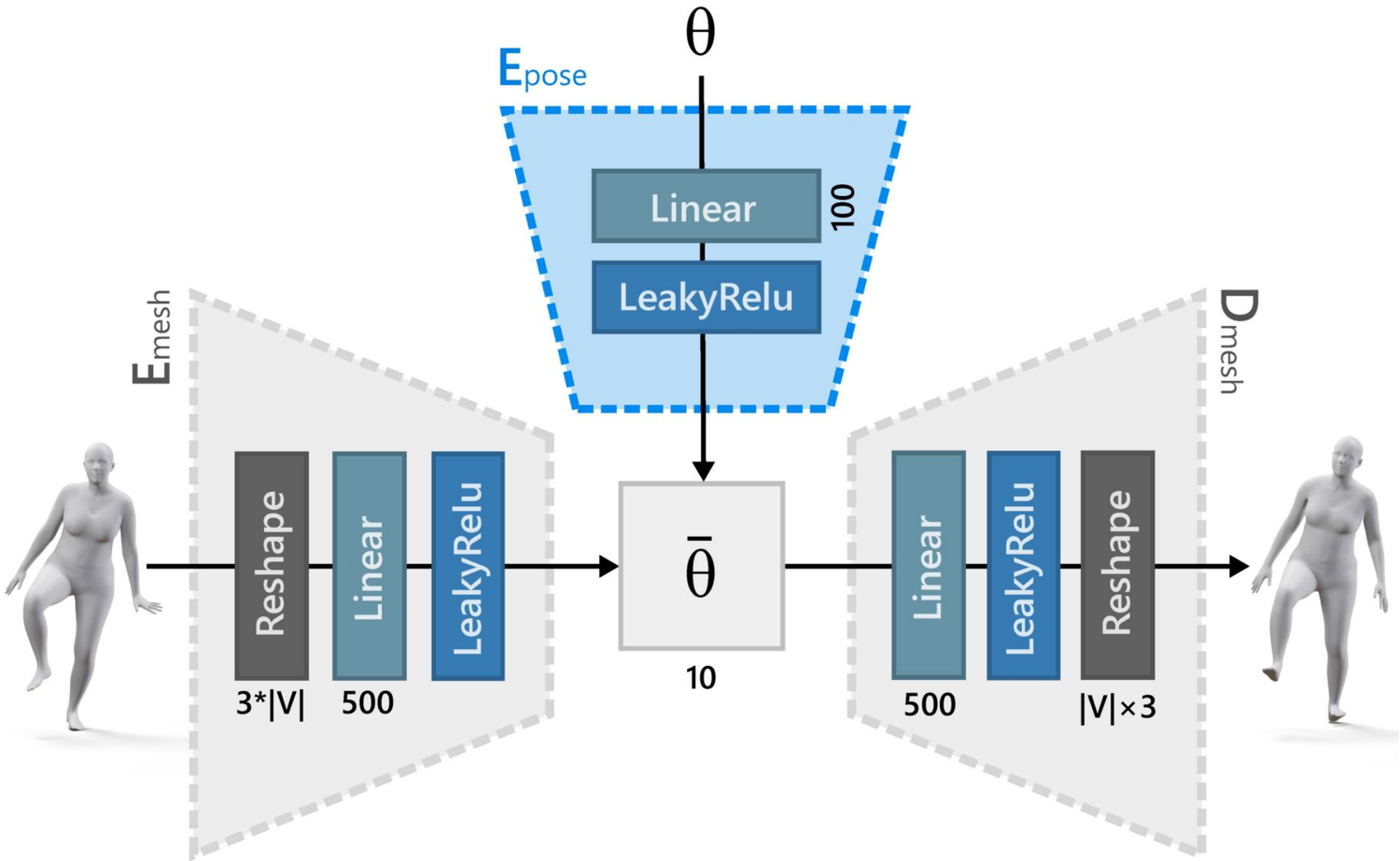
Dimensionality Reduction



Santesteban, I., Garces, E., Otaduy, M. A., & Casas, D. (2020, May). SoftSMPL: data-driven modeling of nonlinear soft-tissue dynamics for parametric humans. In Computer Graphics Forum (Vol. 39, No. 2, pp. 65-75).

Marc Comino
marc.comino@urjc.es

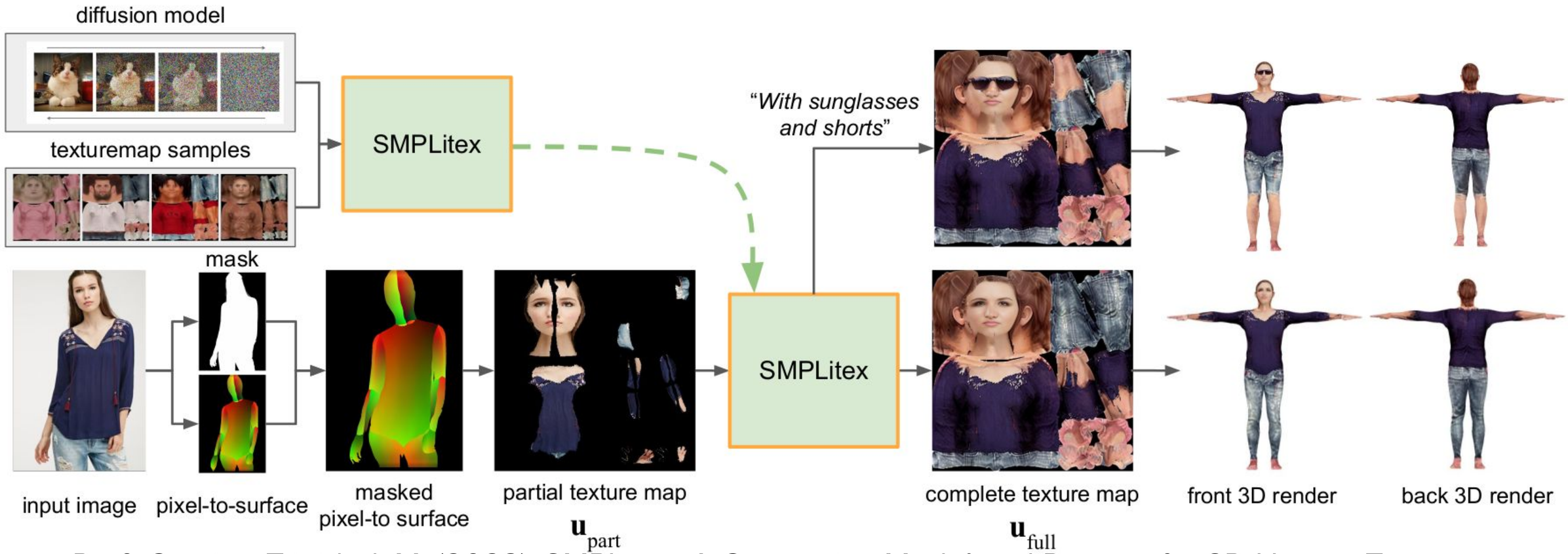
Dimensionality Reduction



Santesteban, I., Garces, E., Otaduy, M. A., & Casas, D. (2020, May). SoftSMPL: data-driven modeling of nonlinear soft-tissue dynamics for parametric humans. In Computer Graphics Forum (Vol. 39, No. 2, pp. 65-75).

Marc Comino
marc.comino@urjc.es

Generative Textures for Avatars



Casas, D., & Comino-Trinidad, M. (2023). SMPLitex: A Generative Model and Dataset for 3D Human Texture Estimation from Single Image.

Marc Comino
marc.comino@urjc.es

Generative Textures for Avatars

"Futuristic astronaut"

"Angela Merkel, shirt"

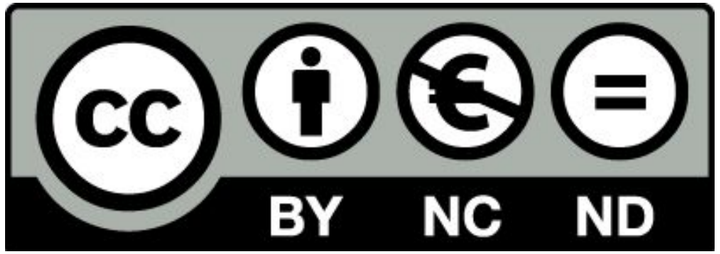
The SMPLitex Dataset

Marc Comino
marc.comino@urjc.es

Thanks!

Q & A

MSLab
Multimodal Simulation Lab



Marc Comino
marc.comino@urjc.es