SMPL made simple FAQs

Brought to you by the researchers at the
Perceiving Systems Department
MPI for Intelligent Systems

SMPL Wiki

Courtesy of Meshcapade

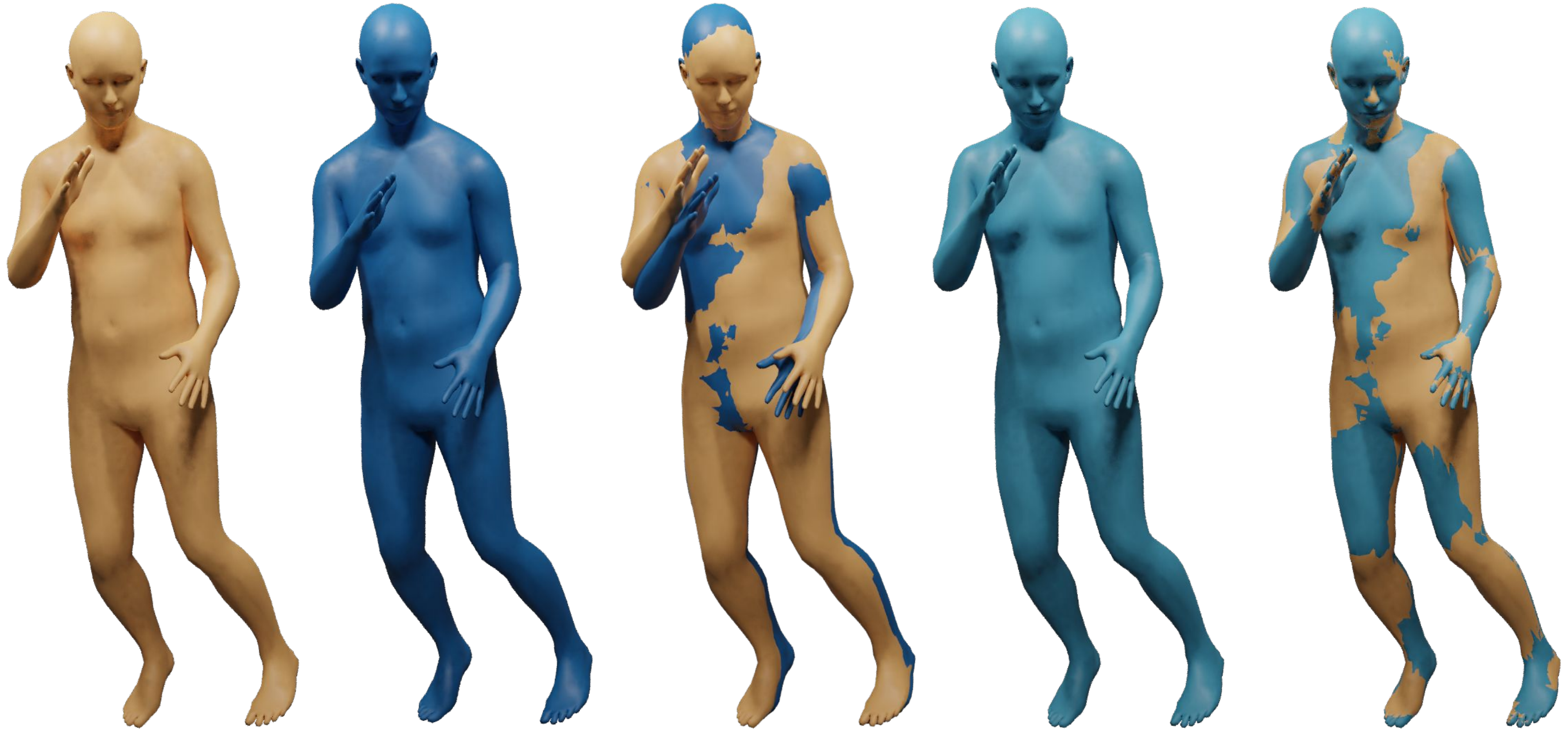
https://meshcapade.wiki/SMPL

Do I need a license for SMPL, etc.?

- SMPL et al. are available free for research purpose
- Commercial use requires a license, which can be easily obtained from meshcapade.com
- SMPL-Model license is needed to *create* bodies
  - This includes the shape space (betas)
- SMPL-Body lets you freely distribute created meshes and poses
  - Like “pdf” for bodies

Vassilis Choutas

How do I convert between SMPL and SMPL-X (or STAR)?



SMPL

SMPL-X from SMPL pose

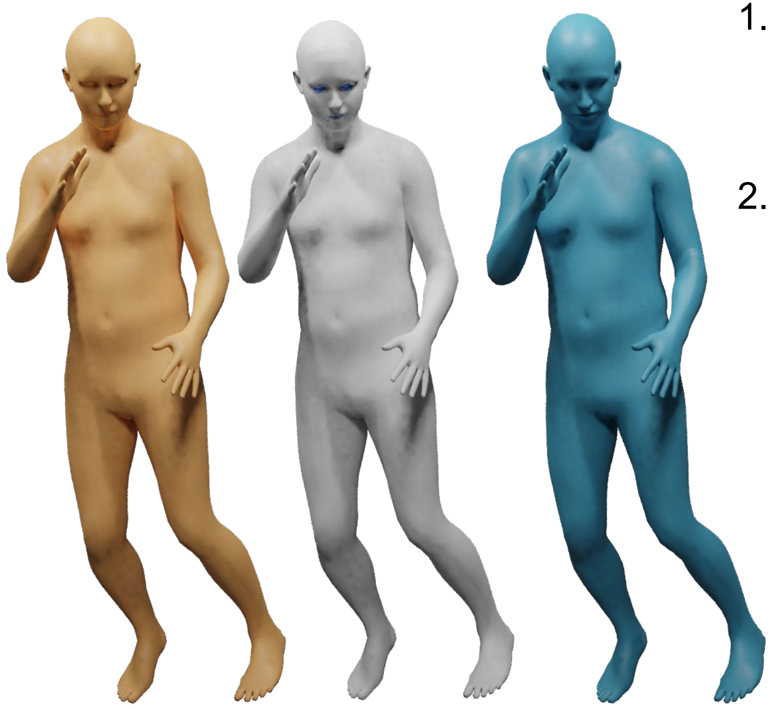
SMPL-X from SMPL pose /SMPL

Correct SMPL-X transfer

Overlay after transfer

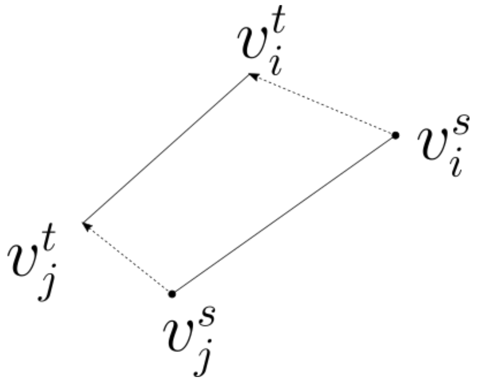
Vassilis Choutas

How do I convert between SMPL and SMPL-X (or STAR)?



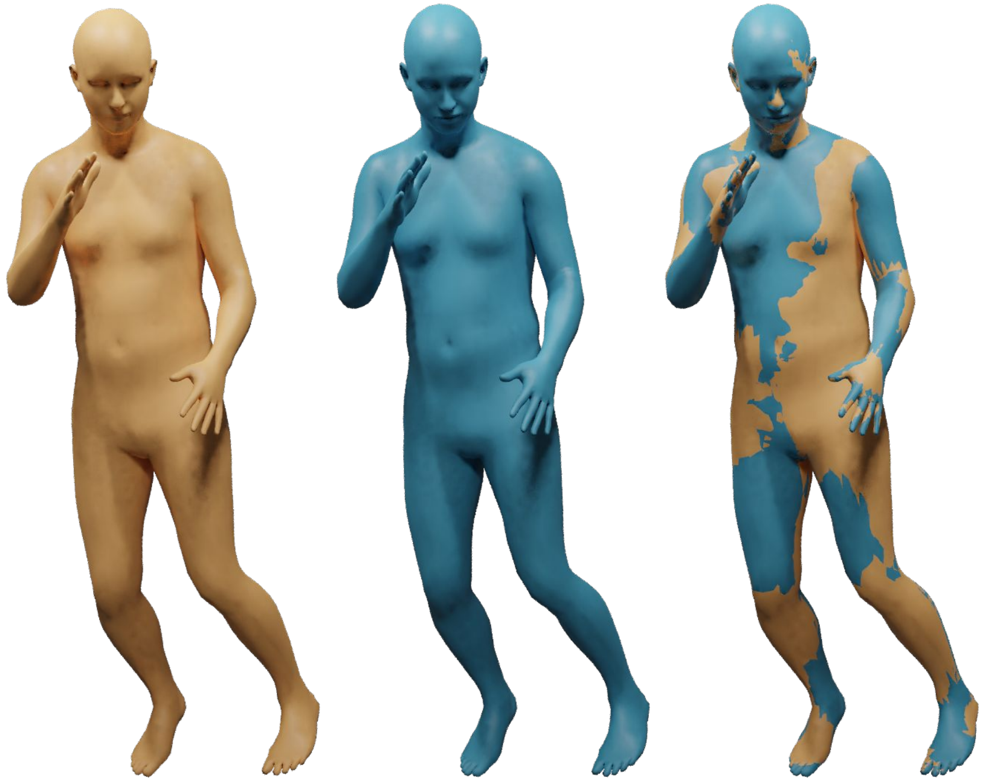
SMPL Deform SMPL to SMPL-X SMPL-X

1. Source to target model topology/triangulation:
   a. Identity for SMPL, SMPL+H and STAR
   b. Triangle and barycentrics for SMPL & SMPL-X + mask to remove invalid matches.
2. Solve for target model parameters
   a. Iterative optimization
   b. $$L_{\mathcal{E}}(\theta, \beta) = \sum_{(i,j)\in\mathcal{E}} \|(v_i^s - v_j^s) - (v_i^t - v_j^t)\|_2^2$$
   c. $$L_V = S_i \|v_i^s - v_i^t\|_2^2$$



Vassilis Choutas

How do I convert between SMPL and SMPL-X (or STAR)?



Code: https://github.com/vchoutas/smplx/tree/master/transfer_model

I want to use AMASS but...

I only want SMPL parameters, not SMPL-H. How do I get these?

SMPL-H is exactly SMPL with the addition of an articulated hand. So the remaining parameters, ignoring the hand, are exactly the same as SMPL (ie betas and thetas). The first 21 joints in SMPL and SMPL-H are identical. So just take the first 21x3 numbers in theta. Ignore the hands by ignoring the joints 22 and above.

But if you want convert AMASS to SMPL-X, you need to use the transfer code in the preceding slides. We will be providing a SMPL-X version of AMASS soon.

What is PA-MPJPE? What is Procrustes?

Problem:

1. Most methods estimate the 3D body pose/shape in the camera coordinate frame not the world.
2. Often the body is tilted relative to the world.
3. Sometimes the body is flipped left to right.
4. This results in high errors even if the pose is roughly right.

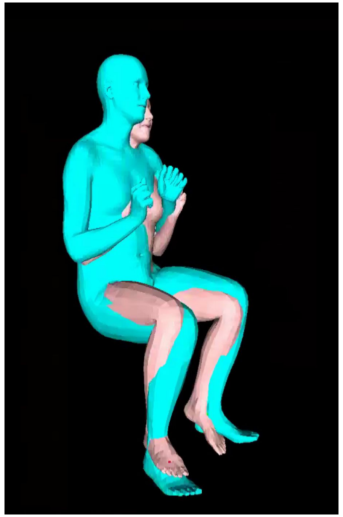
Solution 1:

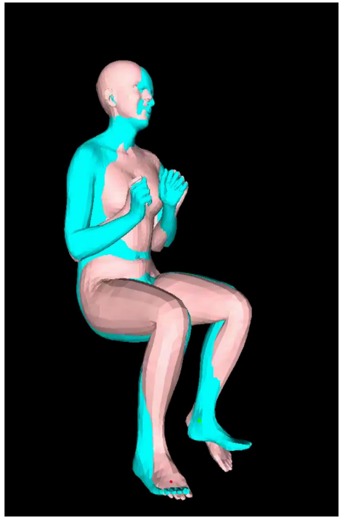
MeanPerJointPositionError (MPJPE) → Mean per joint position error for all joints

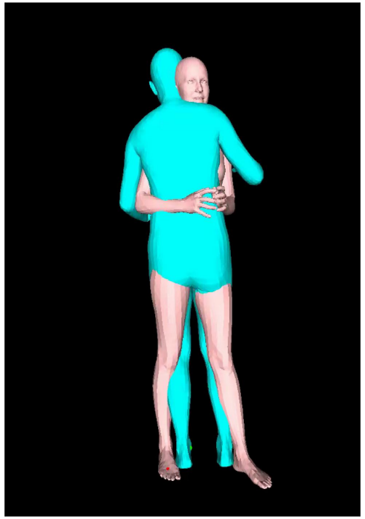
- Calculated after translating the root (‘pelvis’) of estimated body to the groundtruth root.
- This doesn’t get rid of rotation.

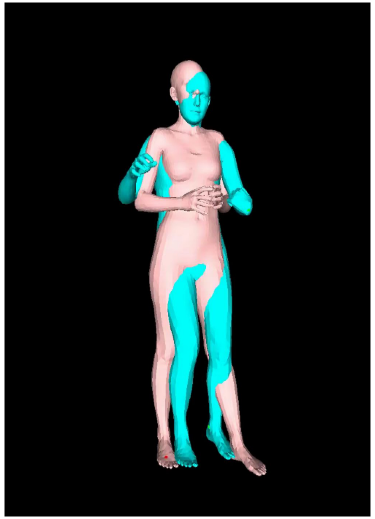
What is PA-MPJPE? What is Procrustes?

Blue - Predictions, Pink - Ground truth (from AGORA https://agora.is.tue.mpg.de/)


Pelvis Aligned


Procrustes Aligned

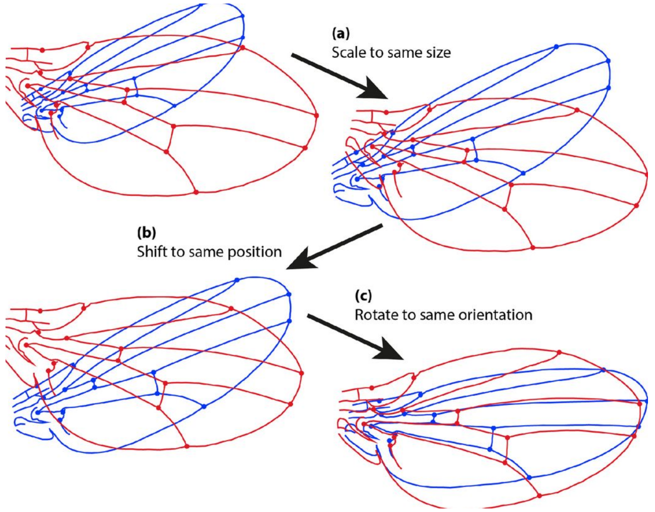

Pelvis Aligned


Procrustes Aligned

What is PA-MPJPE? What is Procrustes?

PA-MPJPE

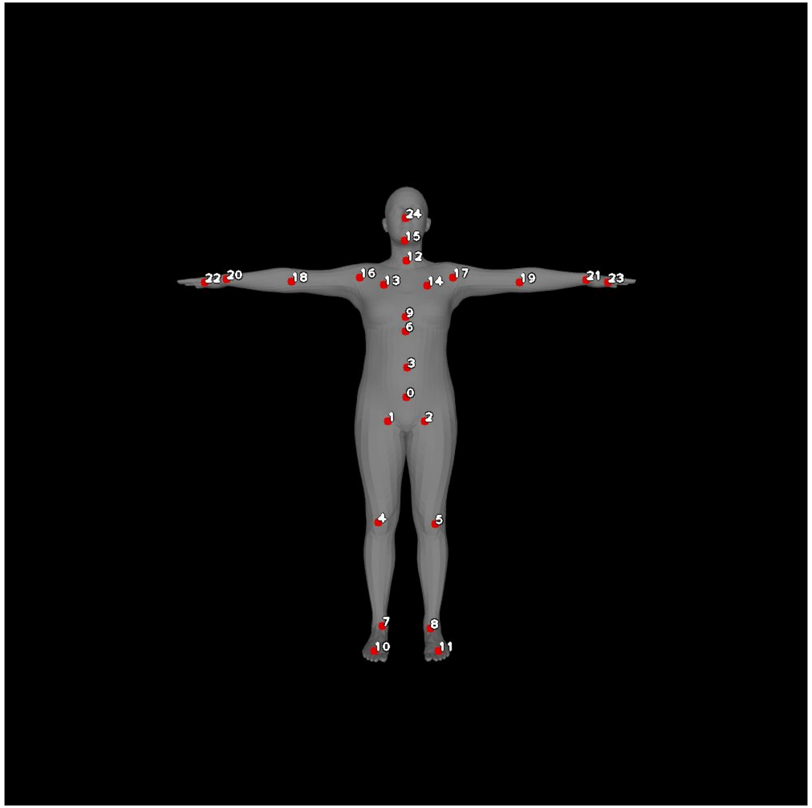
- Uses Procrustes alignment (PA) to solve for translation, scale and rotation between the estimated body and the ground truth.
- Positive: it focuses on pose
- Negative: it hides many sins



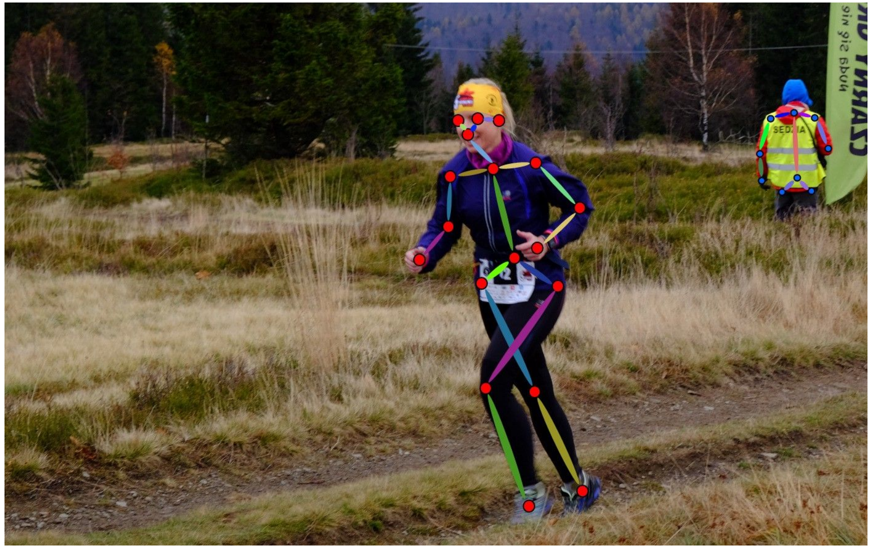
The field should phase out Procrustes. It's a crutch.
We never have the ground truth in real life.
We want algorithms that give the body back in world coordinates.

Michael Black

When do I need to use a joint regressor and why?



SMPL Joints

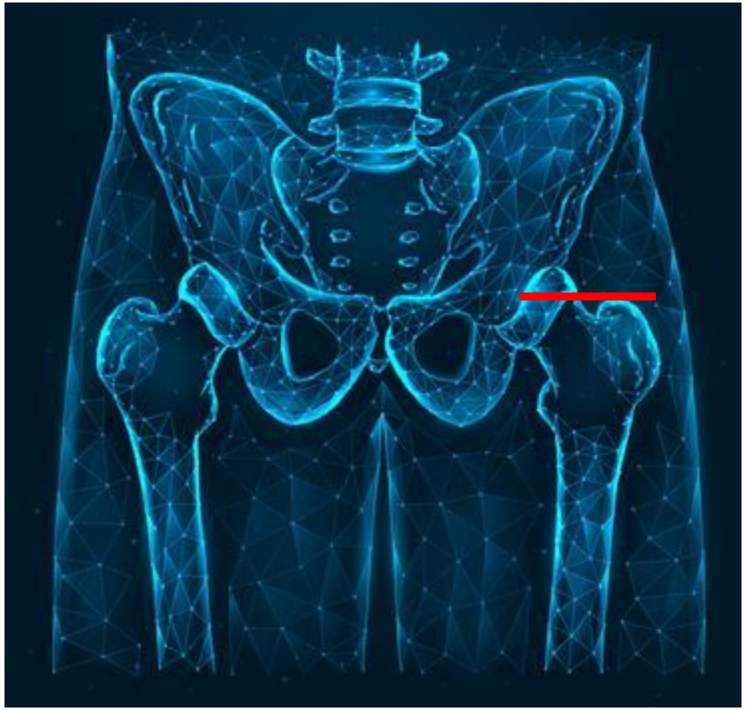


COCO Joints

Human joint labels are not consistent with anatomy. Particularly the hips.
SMPL puts the hip joint where the rotation happens.

Michael Black

Where people label the hip vs where the joint really is



Additionally, every 3D dataset may define the joints and the kinematic tree differently.

Note that the actual joints are never observed.

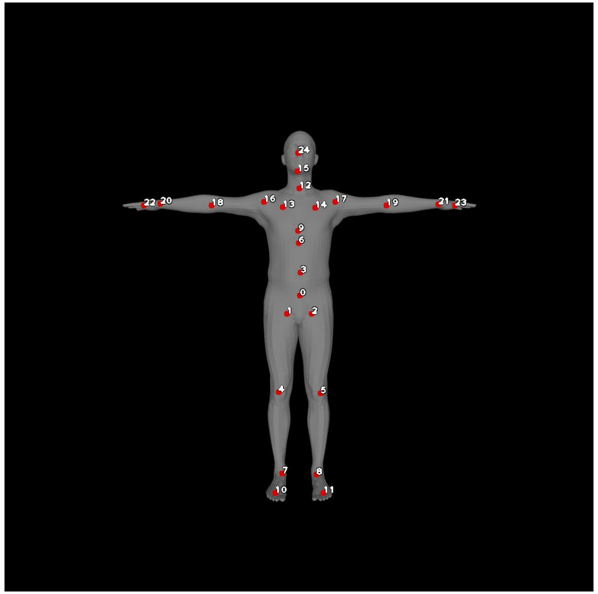
They are always inferred, e.g. by a mocap system.

E.g when using H3.6M, one needs to transform SMPL’s joints to the definition of 3D joints used in the dataset.

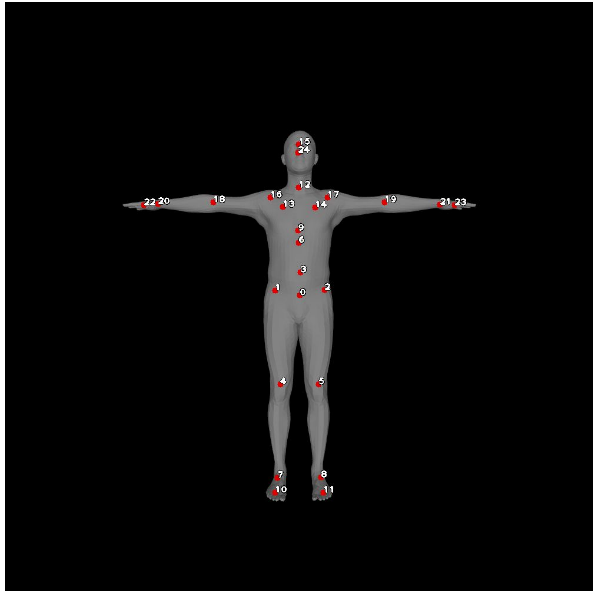
When training with 2D joints from OpenPose, one has to map to 3D joints that project into the image and match in 2D.

Vassilis Choutas

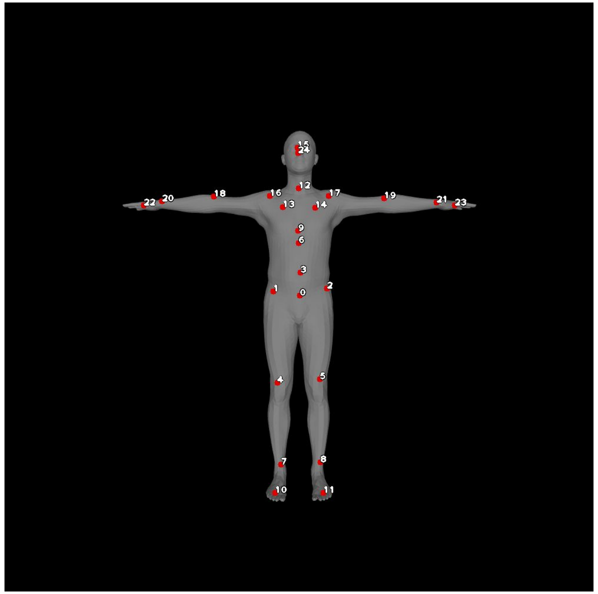
What joint regressor should I use for different datasets?



SMPL joints



HMR J-14
Regressor



SPIN J-14
Regressor

What joint regressor should I use for different datasets?

- Ideally one per dataset
- In practice:
  - Use the same regressor for train and test
  - Fine-tune per dataset, see: https://hassony2.github.io/handobjectconsist.html

PA-MPJPE (mm)	HMR J14 (predictions)	SPIN J14 (predictions)
HMR J14 (GT)	64.1	-
SPIN J14 (GT)	63.62	60.2

- Which quantities should I use for the metrics?
  - 3DPW
    - 24 SMPL joints
    - Vertices
  - AGORA
    - SMPL-X joints
    - Vertices

Naureen Mahmood

I want to use SMPL in Maya without the dependency on numpy.

This is not possible yet. There’s no simpler library to do the array operations.

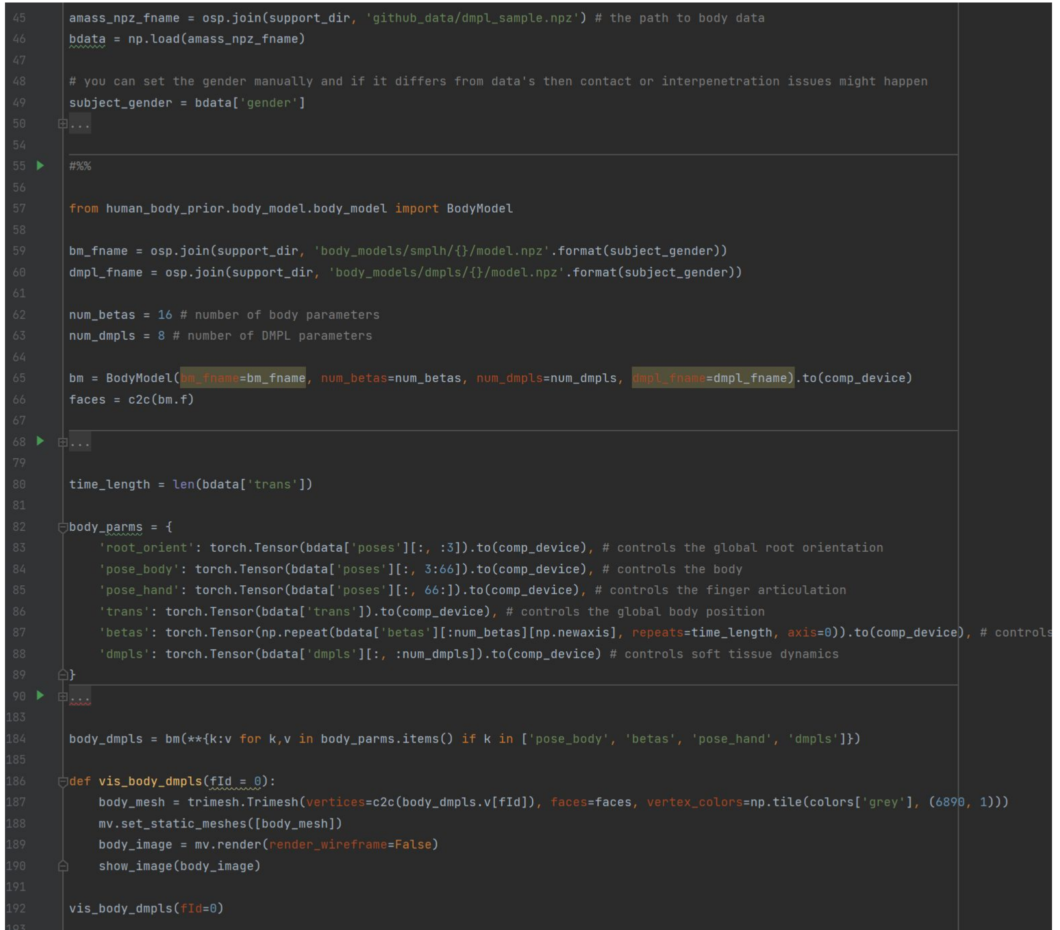
How do I add soft-tissue dynamics to SMPL?

Nima Ghorbani

Animate soft-tissue dynamics with DMPL parameters

Similar to any other parameter of SMPL, e.g. pose, beta

AMASS uses 8 DMPL parameters to extract soft-tissue motions realistically from a sparse set of markers.



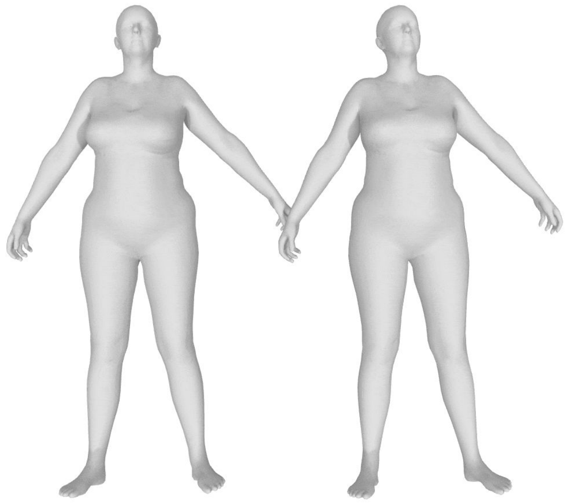
```python
amass_npz_fname = osp.join(support_dir, 'github_data/dmpl_sample.npz') # the path to body data
bdata = np.load(amass_npz_fname)

# you can set the gender manually and if it differs from data's then contact or interpenetration issues might happen
subject_gender = bdata['gender']
...

#%%

from human_body_prior.body_model.body_model import BodyModel

bm_fname = osp.join(support_dir, 'body_models/smplh/{}/model.npz'.format(subject_gender))
dmpl_fname = osp.join(support_dir, 'body_models/dmpls/{}/model.npz'.format(subject_gender))

num_betas = 16 # number of body parameters
num_dmpls = 8 # number of DMPL parameters

bm = BodyModel(bm_fname=bm_fname, num_betas=num_betas, num_dmpls=num_dmpls, dmpl_fname=dmpl_fname).to(comp_device)
faces = c2c(bm.f)

...

time_length = len(bdata['trans'])

body_parms = {
    'root_orient': torch.Tensor(bdata['poses'][:, :3]).to(comp_device), # controls the global root orientation
    'pose_body': torch.Tensor(bdata['poses'][:, 3:66]).to(comp_device), # controls the body
    'pose_hand': torch.Tensor(bdata['poses'][:, 66:]).to(comp_device), # controls the finger articulation
    'trans': torch.Tensor(bdata['trans']).to(comp_device), # controls the global body position
    'betas': torch.Tensor(np.repeat(bdata['betas'][:num_betas][np.newaxis], repeats=time_length, axis=0)).to(comp_device), # controls
    'dmpls': torch.Tensor(bdata['dmpls'][:, :num_dmpls]).to(comp_device) # controls soft tissue dynamics
}
...

body_dmpls = bm(**{k:v for k,v in body_parms.items() if k in ['pose_body', 'betas', 'pose_hand', 'dmpls']})

def vis_body_dmpls(fId = 0):
    body_mesh = trimesh.Trimesh(vertices=c2c(body_dmpls.v[fId]), faces=faces, vertex_colors=np.tile(colors['grey'], (6890, 1)))
    mv.set_static_meshes([body_mesh])
    body_image = mv.render(render_wireframe=False)
    show_image(body_image)

vis_body_dmpls(fId=0)
```

Full Code: https://github.com/nghorbani/amass

Nima Ghorbani

How do I add soft-tissue dynamics to SMPL?

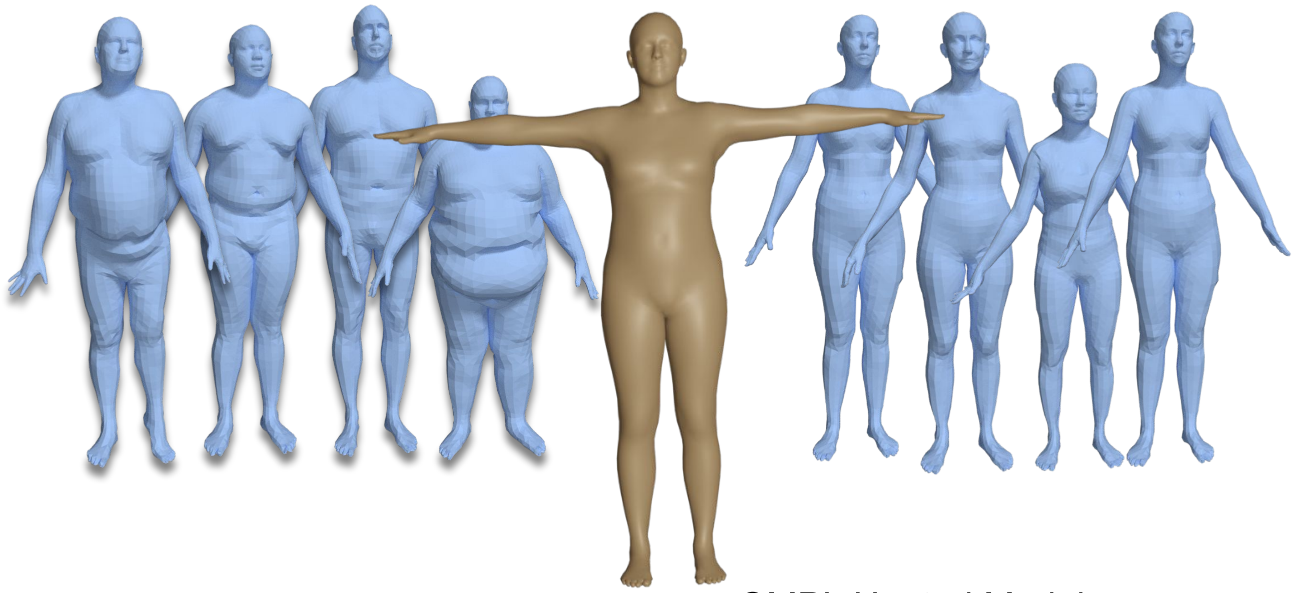
With DMPL



Without DMPL

Full Code: https://github.com/nghorbani/amass

How do I get a gender neutral model?


SMPL Neutral Model

- Neutral models are trained on *male* and *female* subjects.
- Available Neutral Models: **SMPL** , **SMPL+H** , **SMPL-X** and **STAR**
- Check the SMPL Made Simple website (https://smpl-made-simple.is.tue.mpg.de/smplcentral.html)

Ahmed Osman

How do I get SMPL with 300 shape components

Model	Num. Comp.
Male, Female, Neutral **SMPL**	*300*
Male, Female, Neutral **SMPL-X**	*300*
Male, Female, Neutral **STAR**	*300*

- All the models above are currently available with the ‘full shape space’ of 300 shape components.
- Check the SMPL Made Simple website for links: https://smpl-made-simple.is.tue.mpg.de/smplcentral.html

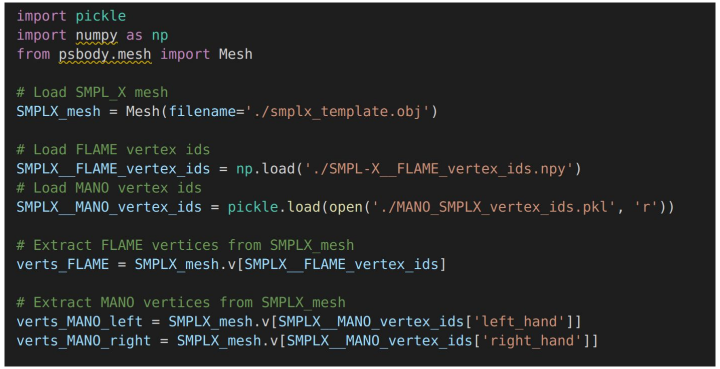
Timo Bolkart

I want just the face (or hand) vertices in SMPL-X

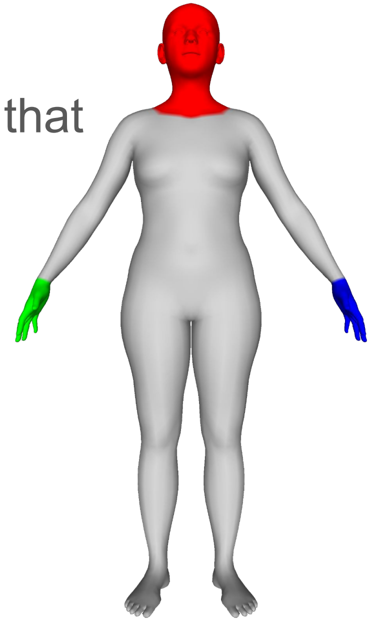
How do you get these?

We provide FLAME and MANO vertex indices of the SMPL-X body as vertex index lists to download at https://smpl-x.is.tue.mpg.de/downloads.

Indexing the SMPL-X body with these index lists returns the vertices that correspond to the MANO and FLAME body parts.



```python
import pickle
import numpy as np
from psbody.mesh import Mesh

# Load SMPL_X mesh
SMPLX_mesh = Mesh(filename='./smplx_template.obj')

# Load FLAME vertex ids
SMPLX__FLAME_vertex_ids = np.load('./SMPL-X__FLAME_vertex_ids.npy')
# Load MANO vertex ids
SMPLX__MANO_vertex_ids = pickle.load(open('./MANO_SMPLX_vertex_ids.pkl', 'r'))

# Extract FLAME vertices from SMPLX_mesh
verts_FLAME = SMPLX_mesh.v[SMPLX__FLAME_vertex_ids]

# Extract MANO vertices from SMPLX_mesh
verts_MANO_left = SMPLX_mesh.v[SMPLX__MANO_vertex_ids['left_hand']]
verts_MANO_right = SMPLX_mesh.v[SMPLX__MANO_vertex_ids['right_hand']]
```



Shashank Tripathi

How do I get the joints out of SMPL?

The easiest way to get joints from SMPL is to use the SMPL class in `body_model.py` in the SMPL-X github repository

https://github.com/vchoutas/smplx/blob/master/smplx/body_models.py#L43

1. Initialize the SMPL constructor object with the correct model path

```
smpl_object = SMPL(model_dir='<PATH_TO_SMPL_PKL>')
```

2. Run forward() function

```
smpl_output = smpl_object.forward(betas, pose, global_orientation)
```

3. Extract joints and vertices

```
joints = smpl_output.joints
vertices = smpl_output.vertices
```

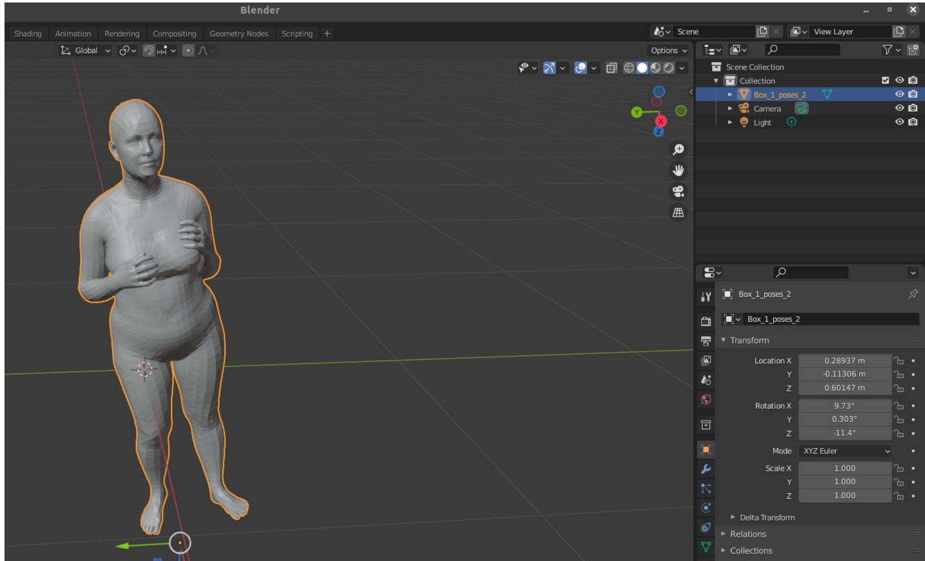
Shashank Tripathi

How to visualize it in Blender?

Blender accepts .obj and .fbx types, so once you get the vertices:

```
import trimesh
mesh = trimesh.Trimesh(vertices=vertices,
                               faces=smpl_output.faces,
                               process=False,
                               maintain_order=True)
mesh_fname = 'my_mesh.obj'
mesh.export(mesh_fname)
```

→ Import the obj file into blender



Box_1_poses_2

Location X 0.28937 m
Y -0.11306 m
Z 0.60147 m
Rotation X 9.73°
Y 0.303°
Z -11.4°
Mode XYZ Euler
Scale X 1.000
Y 1.000
Z 1.000

Shashank Tripathi

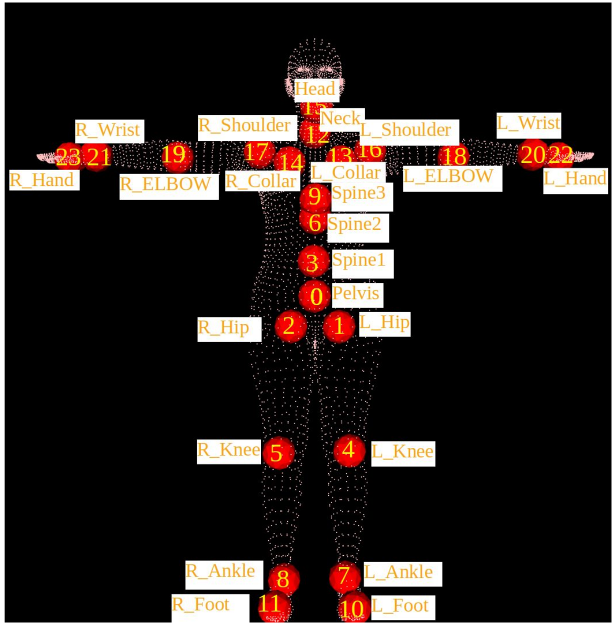
I want to create a skeleton.

How do I get the kinematic tree and joints?

A. The following is the mapping between SMPL joint ids to joint names

```
'Pelvis',     # 0
'L_Hip',      # 1
'R_Hip',      # 2
'Spine1',     # 3
'L_Knee',     # 4
'R_Knee',     # 5
'Spine2',     # 6
'L_Ankle',    # 7
'R_Ankle',    # 8
'Spine3',     # 9
'L_Foot',     # 10
'R_Foot',     # 11
```

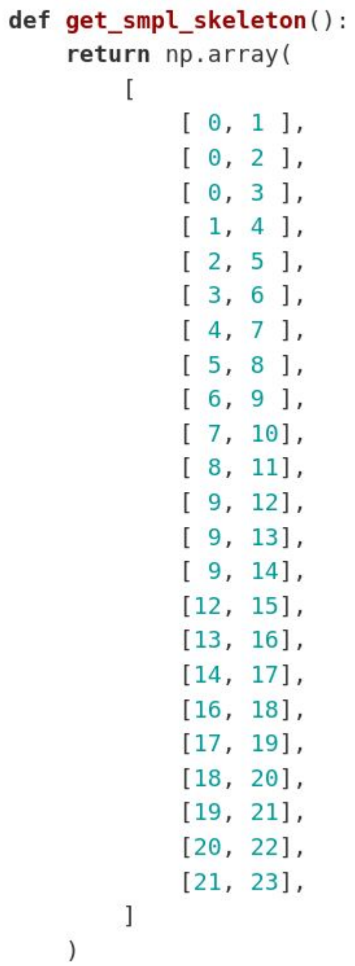
```
'Neck',        # 12
'L_Collar',    # 13
'R_Collar',    # 14
'Head',        # 15
'L_Shoulder',# 16
'R_Shoulder',# 17
'L_Elbow',     # 18
'R_Elbow',     # 19
'L_Wrist',     # 20
'R_Wrist',     # 21
'L_Hand',      # 22
'R_Hand',      # 23
```



Shashank Tripathi

Kinematic tree

And the connectivity map

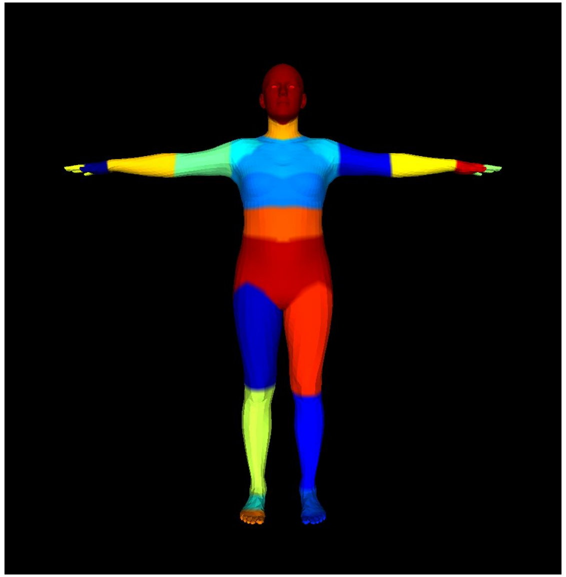


```
def get_smpl_skeleton():
    return np.array(
        [
            [ 0, 1 ],
            [ 0, 2 ],
            [ 0, 3 ],
            [ 1, 4 ],
            [ 2, 5 ],
            [ 3, 6 ],
            [ 4, 7 ],
            [ 5, 8 ],
            [ 6, 9 ],
            [ 7, 10],
            [ 8, 11],
            [ 9, 12],
            [ 9, 13],
            [ 9, 14],
            [12, 15],
            [13, 16],
            [14, 17],
            [16, 18],
            [17, 19],
            [18, 20],
            [19, 21],
            [20, 22],
            [21, 23],
        ]
    )
```

Muhammed Kocabas

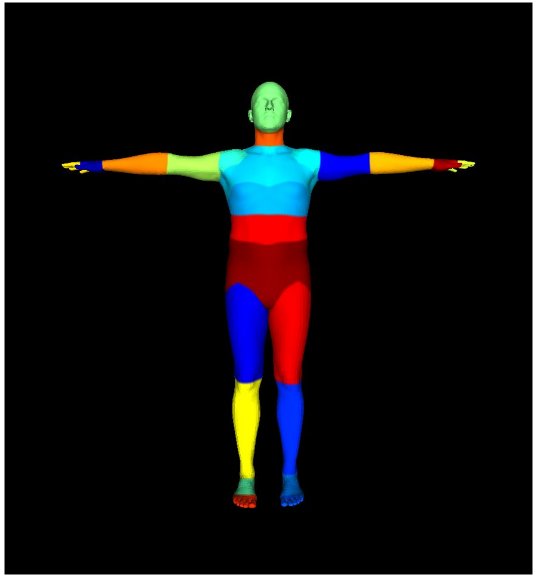
I want the body segmented into parts.

SMPL-X



- Per-vertex body part labels:
  https://github.com/Meshcapade/wiki/tree/main/assets/SMPL_body_segmentation/smplx

SMPL



- Per-vertex body part labels:
  https://github.com/Meshcapade/wiki/tree/main/assets/SMPL_body_segmentation/smpl
- Code to generate above images:
  https://gist.github.com/mkocabas/238faa8a24c44d60fb30ffce367f9ec7

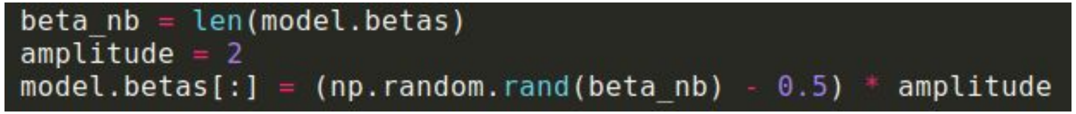
How do I sample body shapes?

- Shape space defined by PCA (Principal Component Analysis)
  - Shape vector $model.betas = [\beta_0, \ ..., \ \beta_{N-1}]$
  - Base vectors $B_i = model.shapedirs[:, :, i]$
  - A body shape can be written:

$$x = M + \sum_i \beta_i \overrightarrow{B_i} \qquad \left\| \overrightarrow{B_i} \right\| = \sigma_i$$

Note: PCA returns vectors B_i of unit norm. For convenience we scale them by sigma. So to sample according to the Gaussian over body shapes, you can just sample a Gaussian where the covariance matrix is the identity.

- To sample a shape
  - ***β_i* in range [-2, 2]**
  - ⇒ shape in [-2σ, 2σ], where ~95% of the training subjects lie



```
beta_nb = len(model.betas)
amplitude = 2
model.betas[:] = (np.random.rand(beta_nb) - 0.5) * amplitude
```

Marilyn Keller

How do I use the shape prior as a loss?

- When optimizing for the shape parameters β
  - Keep plausible shapes

- L2 loss to regularize the shape vector
  - Scale each dimension by its corresponding variance

$$L(\beta) = \sum_i \sigma_i^2 \beta_i^2$$

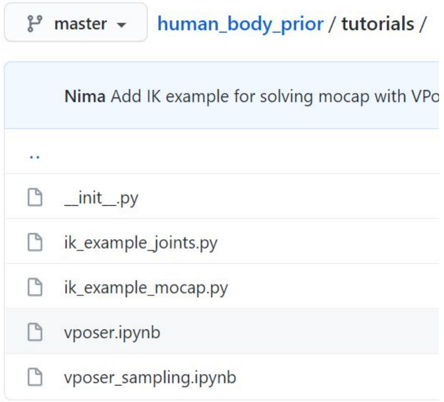
Note: In practice most people just put an L2 loss on the betas.

I need a pose prior for my application

VPoser:

- SMPL body pose prior as latent code of **variational autoencoder**
- Trained on **AMASS**
- End-to-end **differentiable**
- Provides a way to penalize impossible poses while admitting valid ones
- Effectively models **correlations** among the joints of the body
- Can be used to **generate** valid 3D human poses for data-dependent tasks
- Enables **inverse kinematic** in batch mode without requiring initialization

VPoser encoding/decoder



master human_body_prior / tutorials /

Nima Add IK example for solving mocap with VPo

..

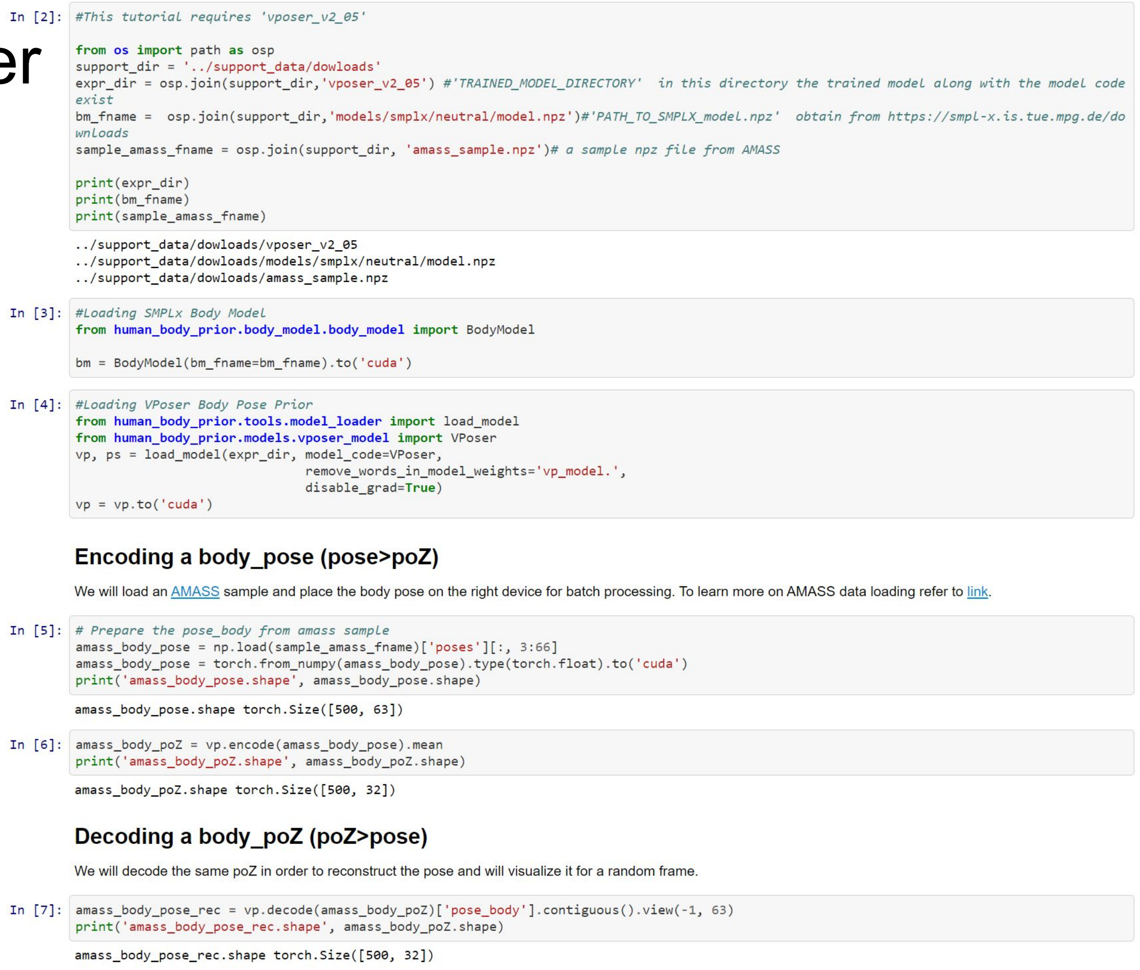
__init__.py

ik_example_joints.py

ik_example_mocap.py

vposer.ipynb

vposer_sampling.ipynb



```
In [2]: #This tutorial requires 'vposer_v2_05'

        from os import path as osp
        support_dir = '../support_data/dowloads'
        expr_dir = osp.join(support_dir,'vposer_v2_05') #'TRAINED_MODEL_DIRECTORY'  in this directory the trained model along with the model code exist
        bm_fname =  osp.join(support_dir,'models/smplx/neutral/model.npz')#'PATH_TO_SMPLX_model.npz'  obtain from https://smpl-x.is.tue.mpg.de/downloads
        sample_amass_fname = osp.join(support_dir, 'amass_sample.npz')# a sample npz file from AMASS

        print(expr_dir)
        print(bm_fname)
        print(sample_amass_fname)

        ../support_data/dowloads/vposer_v2_05
        ../support_data/dowloads/models/smplx/neutral/model.npz
        ../support_data/dowloads/amass_sample.npz

In [3]: #Loading SMPLx Body Model
        from human_body_prior.body_model.body_model import BodyModel

        bm = BodyModel(bm_fname=bm_fname).to('cuda')

In [4]: #Loading VPoser Body Pose Prior
        from human_body_prior.tools.model_loader import load_model
        from human_body_prior.models.vposer_model import VPoser
        vp, ps = load_model(expr_dir, model_code=VPoser,
                                      remove_words_in_model_weights='vp_model.',
                                      disable_grad=True)
        vp = vp.to('cuda')
```

Encoding a body_pose (pose>poZ)

We will load an AMASS sample and place the body pose on the right device for batch processing. To learn more on AMASS data loading refer to link.

```
In [5]: # Prepare the pose_body from amass sample
        amass_body_pose = np.load(sample_amass_fname)['poses'][:, 3:66]
        amass_body_pose = torch.from_numpy(amass_body_pose).type(torch.float).to('cuda')
        print('amass_body_pose.shape', amass_body_pose.shape)

        amass_body_pose.shape torch.Size([500, 63])

In [6]: amass_body_poZ = vp.encode(amass_body_pose).mean
        print('amass_body_poZ.shape', amass_body_poZ.shape)

        amass_body_poZ.shape torch.Size([500, 32])
```

Decoding a body_poZ (poZ>pose)

We will decode the same poZ in order to reconstruct the pose and will visualize it for a random frame.

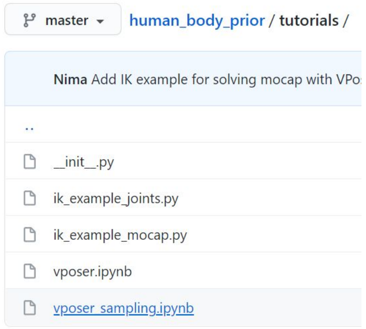
```
In [7]: amass_body_pose_rec = vp.decode(amass_body_poZ)['pose_body'].contiguous().view(-1, 63)
        print('amass_body_pose_rec.shape', amass_body_poZ.shape)

        amass_body_pose_rec.shape torch.Size([500, 32])
```

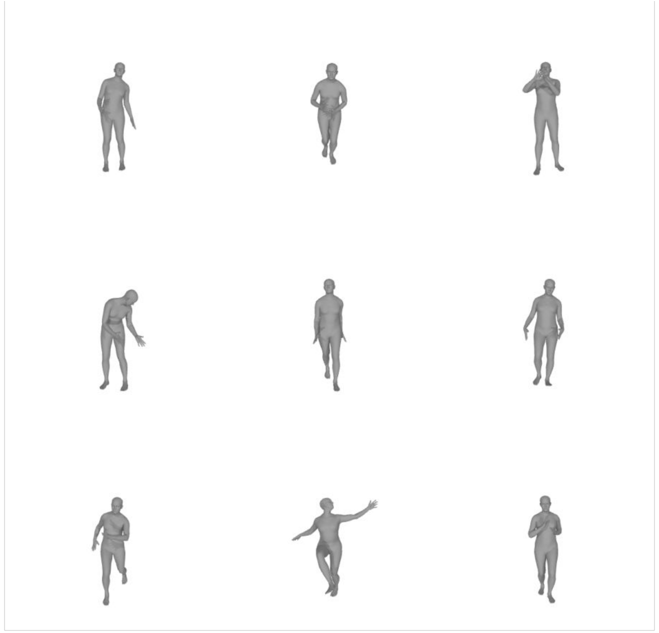
Full Code: https://github.com/nghorbani/human_body_prior

Nima Ghorbani

VPoser Sample New Poses

```
num_poses = 9 # number of body poses in each batch

sampled_pose_body = vp.sample_poses(num_poses=num_poses)['pose_body'].contiguous().view(num_poses, -1) # will a generate Nx1x21x3 tensor
of body poses
images = render_smpl_params(bm, {'pose_body':sampled_pose_body}).reshape(3,3,1,400,400,3)
img = imagearray2file(images)
show_image(np.array(img[0]))
```


master
human_body_prior / tutorials /
Nima Add IK example for solving mocap with VPo
..
__init__.py
ik_example_joints.py
ik_example_mocap.py
vposer.ipynb
vposer_sampling.ipynb

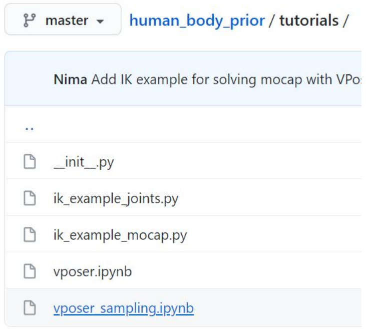


Full Code: https://github.com/nghorbani/human_body_prior

Nima Ghorbani

VPoser For Interpolating Between Poses

$$poZ_{inp} = \alpha\, poZ_1 + (1-\alpha)\, poZ_2.$$



master human_body_prior / tutorials /

Nima Add IK example for solving mocap with VPo

..

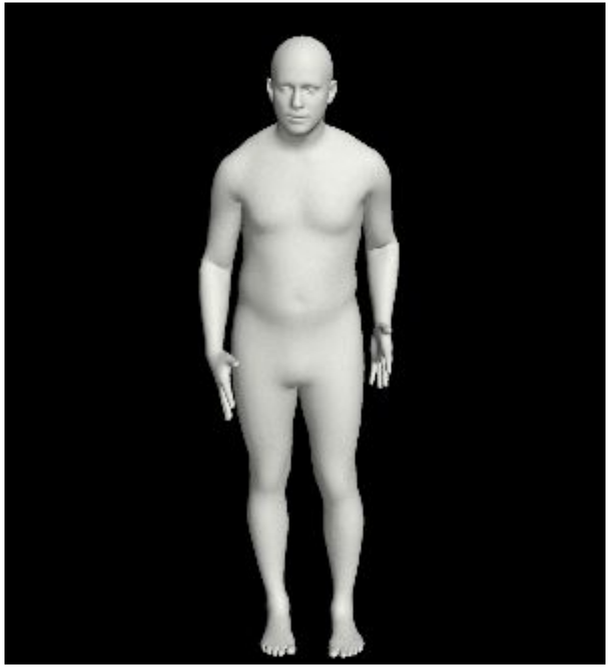
__init__.py

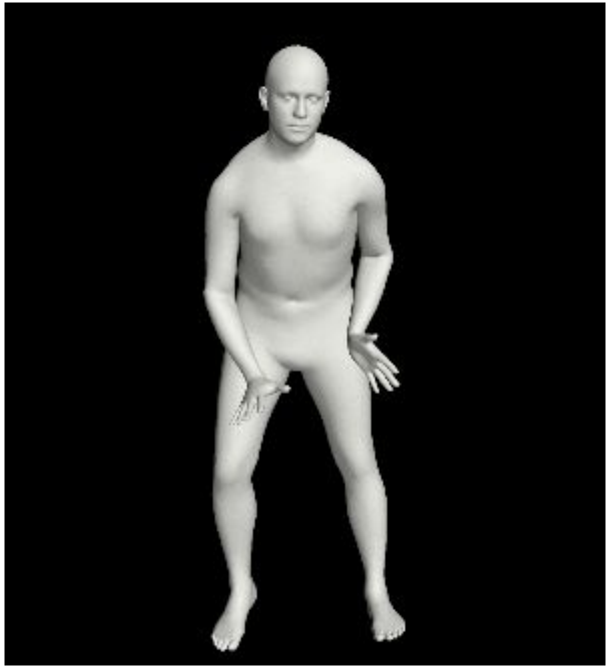
ik_example_joints.py

ik_example_mocap.py

vposer.ipynb

vposer_sampling.ipynb





Full Code: https://github.com/nghorbani/human_body_prior

Omid Taheri

How do I convert SMPL part rotations to other formats

Different rotation representations:

1. Axis-Angle (N, 3) → SMPL’s default input
2. Rotation Matrix (N, 3, 3) → What SMPL uses behind the scene
3. Quaternion (N, 4)
4. Euler Angles (N, 3)
5. 6D Rotation [1] (N, 3, 2)
    - By dropping the last row of the rotation Matrix representation

N → Batch size

[1] Zhou, Y., Barnes, C., Lu, J., Yang, J., & Li, H. On the Continuity of Rotation Representations in Neural Networks. IEEE Conference on Computer Vision and Pattern Recognition, 2019.

Omid Taheri

How do I convert SMPL part rotations to other formats

To convert rotations to each other

- `pytorch3d` package here.
- `kaolin`
- `torchgeometry`

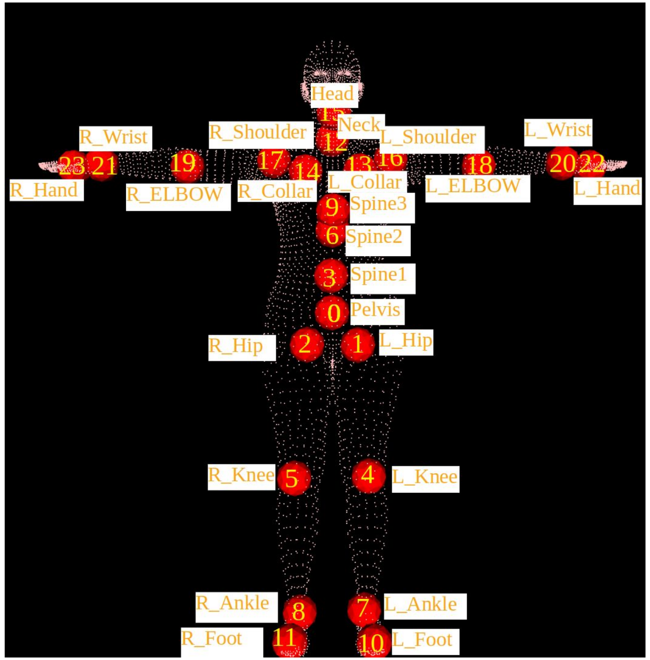
```python
from pytorch3d import transforms

# axis angle
aa   = transforms.matrix_to_axis_angle(rot_mat)
aa   = transforms.quaternion_to_axis_angle(quats)

# rotation matrix
rot_mat = transforms.rotation_6d_to_matrix(d6)
rot_mat = transforms.quaternion_to_matrix(quat)
rot_mat = transforms.euler_angles_to_matrix(euler, convention)
rot_mat = transforms.axis_angle_to_matrix(aa)

# quaternions

quats = transforms.matrix_to_quaternion(rot_mat)
quats = transforms.axis_angle_to_quaternion(aa)

# 6d
d6 = transforms.matrix_to_rotation_6d(rot_mat)

# Euler angles
euler = transforms.matrix_to_euler_angles(rot_mat, convention)
```

Omid Taheri

How do I get parameters for the forward pass of SMPL/SMPLX from full-pose

SMPL full pose in axis-angle → (N,72)

```
#pose is the SMPL full-pose in axis angle representation

global_orient = pose[:, :3]
body_pose = pose[:, 3:]

body_parms = {
'Global_orient':global_orient,
'body_pose': body_pose,
'transl': trans
}

smpl_model.forward(**body_params)
```



SMPL joint ids

Omid Taheri

How do I get parameters for the forward pass of SMPL/SMPLX from full-pose

SMPL-X full pose in axis-angle → (N,165)

```
#pose is the SMPL fullpose in axis angle representation

global_orient = pose[:, :3]
body_pose = pose[:, 3:66]
jaw_pose  = pose[:, 66:69]
leye_pose = pose[:, 69:72]
reye_pose = pose[:, 72:75]
left_hand_pose = pose[:, 75:120]
right_hand_pose = pose[:, 120:]

body_parms = {
'global_orient': global_orient,'body_pose': body_pose,
'jaw_pose': jaw_pose,'leye_pose': leye_pose,
'reye_pose': reye_pose,'left_hand_pose': left_hand_pose,
'right_hand_pose': right_hand_pose,'transl': trans}

smplx_model.forward(**body_params)
```

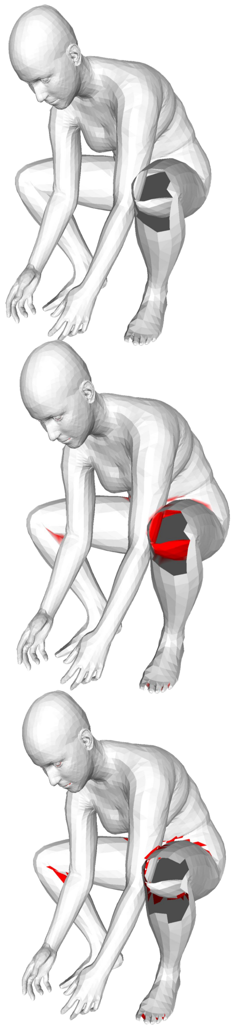
Lea Müller

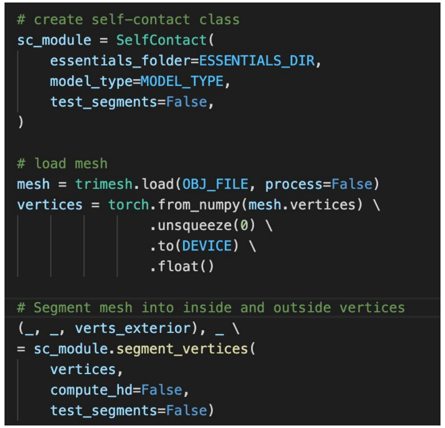
I want to test for interpenetration

How do I do it quickly? This is a memory or time intensive operation.

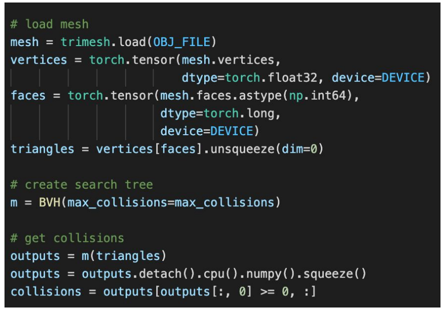
Winding numbers: **Find intersecting <u>vertices</u>**
- Jacobson et al., "Robust inside-outside segmentation using generalized winding numbers", ACM Trans. Graph. 32, July 2013
- Re-implementation (used TUCH and SMPLify-XMC): https://github.com/muelea/selfcontact

Mesh intersection test: **Find intersecting <u>triangles</u>** (FASTER)
- Ballan et al., "Motion capture of hands in action using discriminative salient points" ECCV '12
- Karras et al., "Fast Parallel Construction of High-Quality Bounding Volume Hierarchies, Proceedings of the 5th High-Performance Graphics Conference
- Re-implementation (used in SMPLify-X): https://github.com/vchoutas/torch-mesh-isect





```python
# create self-contact class
sc_module = SelfContact(
    essentials_folder=ESSENTIALS_DIR,
    model_type=MODEL_TYPE,
    test_segments=False,
)

# load mesh
mesh = trimesh.load(OBJ_FILE, process=False)
vertices = torch.from_numpy(mesh.vertices) \
                .unsqueeze(0) \
                .to(DEVICE) \
                .float()

# Segment mesh into inside and outside vertices
(_, _, verts_exterior), _ \
= sc_module.segment_vertices(
    vertices,
    compute_hd=False,
    test_segments=False)
```



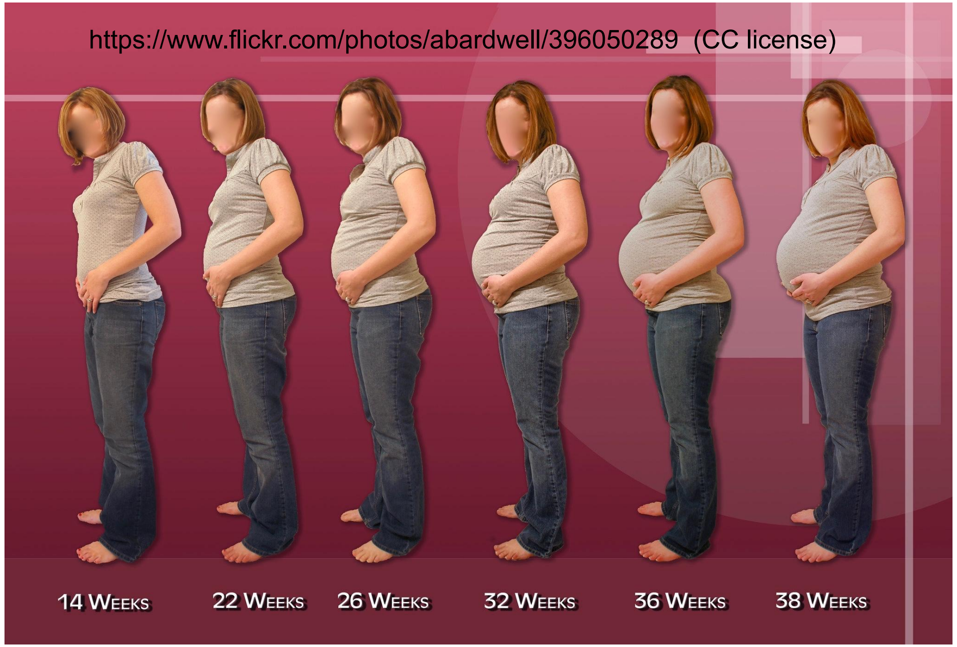
```python
# load mesh
mesh = trimesh.load(OBJ_FILE)
vertices = torch.tensor(mesh.vertices,
                        dtype=torch.float32, device=DEVICE)
faces = torch.tensor(mesh.faces.astype(np.int64),
                     dtype=torch.long,
                     device=DEVICE)
triangles = vertices[faces].unsqueeze(dim=0)

# create search tree
m = BVH(max_collisions=max_collisions)

# get collisions
outputs = m(triangles)
outputs = outputs.detach().cpu().numpy().squeeze()
collisions = outputs[outputs[:, 0] >= 0, :]
```

Michael Black

I ran SMPLify (HMR) and the body shape is not right.

This is natural; estimating a full-3D shape from only a sparse set of 2D joints is highly ambiguous.

Think of a pregnant woman; her shape changes drastically, while her skeletal joints stay nearly the same.



https://www.flickr.com/photos/abardwell/396050289 (CC license)
14 WEEKS
22 WEEKS
26 WEEKS
32 WEEKS
36 WEEKS
38 WEEKS

To estimate 3D shape, we need information beyond 2D joints.

SMPL is limited to the distribution of the training data.

It can’t represent babies, body builders, amputees, etc.

Priyanka Patel

Is there a library of example SMPL (SMPL-X) poses?

One could use AGORA SMPL-X/SMPL pkl files to get the sample poses. To get the pkl files:

1. Go to https://agora.is.tue.mpg.de
2. Downloads -> Ground Truth Fittings -> SMPL-X/SMPL fits

AMASS (in SMPL) -- https://amass.is.tue.mpg.de

TUCH contact poses (in SMPL-X) -- https://tuch.is.tue.mpg.de

Can I get the SMPL training data?

Sorry, no.

The SMPL shape space is trained from CAESAR, which we cannot distribute.

STAR uses CAESAR and SizeUSA and the same applies.

Our pose dataset is also not available due to human subjects limitations.

But the FAUST dataset does provide registered poses.

Michael Black

How can I train my own SMPL model?

Currently we do not have training code online.

Good (well registered) training data is key. This is very hard to produce.

It is not so easy to train a SMPL model from scratch.

Curating the data, evaluating intermediate models, fixing problems, adding more data, etc. is all necessary to avoid spurious long-range correlations and artifacts.

Michael Black

Dataset evaluation questions

Is it possible to evaluate mesh estimation on the MuPoTS dataset? Most of the methods only evaluate mesh estimation on 3DPW and MPII-3DHP, including VIBE, why is that?

How do we unify different skeleton models from different 3D datasets, H36M, 3DHP, 3DPW, MuPoTS, etc?

Is it possible to make a single benchmark by including all these datasets?

MuPoTS, H3.6M, MPII-3DHP all provide 3D joints, not full 3D surfaces or SMPL parameters.

Evaluation datasets with SMPL or SMPL-X ground truth include 3DPW and AGORA.

We have SMPL fits to H3.6M but are not permitted to release them.

Joachim Tesch

I want to pose SMPL myself. How can I do it?

Blender

- Use the SMPL/SMPL-X Blender add-on to bring model into Blender
- Use Blender pose tools to pose model
  - Use Blender add-on to export current full-body pose in Rodrigues format to console for later use in Python code
  - Use Blender add-on to auto-calculate pose corrective weights for current pose
- See also: SMPL-X application integrations presentation in this tutorial

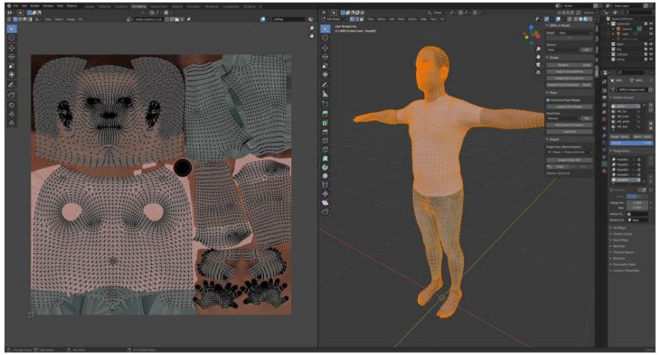
Joachim Tesch

UV maps and textures

How do they work?

- UV maps provide a mapping of each 3D mesh vertex into 2D image space by unwrapping the 3D mesh surface. This technique allows then to map image color information onto the 3D model at a high quality even on low-resolution meshes.

Where can I get textures? What do I need to know about these?



- SMPL-X sample textures are included in the SMPL-X Blender add-on
  - Female/Male, 4096x4096 PNG
  - **Compatible with SMPL UV map**
- Provided by Meshcapade
  - Creative Commons non-commercial license (CC BY-NC 4.0)
  - modify and redistribute

Do you have UV maps for SMPL, SMPL-X, etc? Faces and UV maps?

- UV maps for SMPL and SMPL-X are provided on the SMPL/SMPL-X websites
  - Blender SMPL-X add-on already has UV maps setup
- UV maps are identical between male/female/neutral models so you can easily swap textures

Joachim Tesch

How do I generate animations in FBX format?

Blender workflow

- Use the SMPL-X for Blender add-on
- Keyframe the desired motions with Blender pose tools
- For each individual frame keyframe pose corrective weights using the SMPL-X Blender add-on functionality
  - This can be automated with Blender Python API.
  - Future versions of the SMPL-X Blender add-on will help automating this process.
- Export to FBX