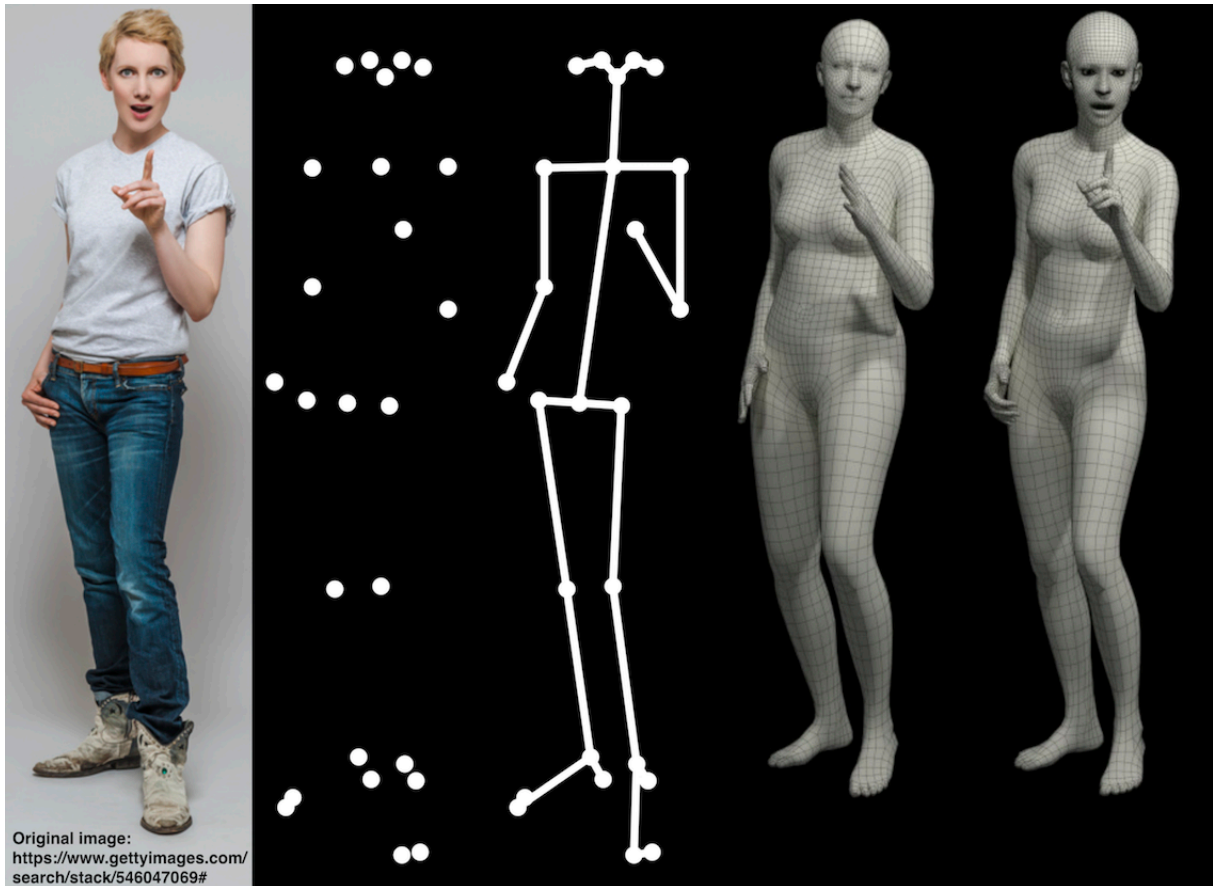


---

## Expressive Body Capture: 3D Hands, Face, and Body from a Single Image

[Project Page] [Paper] [Supp. Mat.]



### Table of Contents

- License
- Description
  - Fitting
  - Different Body Models
  - Visualizing Results
- Dependencies
- Citation
- Acknowledgments
- Contact

---

## License

Software Copyright License for **non-commercial scientific research purposes**. Please read carefully the terms and conditions and any accompanying documentation before you download and/or use the SMPL-X/SMPLify-X model, data and software, (the “Model & Software”), including 3D meshes, blend weights, blend shapes, textures, software, scripts, and animations. By downloading and/or using the Model & Software (including downloading, cloning, installing, and any other use of this github repository), you acknowledge that you have read these terms and conditions, understand them, and agree to be bound by them. If you do not agree with these terms and conditions, you must not download and/or use the Model & Software. Any infringement of the terms of this agreement will automatically terminate your rights under this License.

## Disclaimer

The original images used for the figures 1 and 2 of the paper can be found in this link. The images in the paper are used under license from gettyimages.com. We have acquired the right to use them in the publication, but redistribution is not allowed. Please follow the instructions on the given link to acquire right of usage. Our results are obtained on the  $483 \times 724$  pixels resolution of the original images.

## Description

This repository contains the fitting code used for the experiments in Expressive Body Capture: 3D Hands, Face, and Body from a Single Image.

## Fitting

Run the following command to execute the code:

```
1 python smplifyx/main.py --config cfg_files/fit_smplx.yaml
2   --data_folder DATA_FOLDER
3   --output_folder OUTPUT_FOLDER
4   --visualize="True/False"
5   --model_folder MODEL_FOLDER
6   --vposer_ckpt VPOSER_FOLDER
7   --part_segm_fn smplx_parts_segm.pkl
```

where the `DATA_FOLDER` should contain two subfolders, *images*, where the images are located, and *keypoints*, where the OpenPose output should be stored.

---

## Different Body Models

To fit SMPL or SMPL+H, replace the *yaml* configuration file with either *fit\_smpl.yaml* or *fit\_smplx.yaml*, i.e.: \* for SMPL: `Shell python smplifyx/main.py --config cfg_files/fit_smpl.yaml --data_folder DATA_FOLDER --output_folder OUTPUT_FOLDER --visualize="True/False"--model_folder MODEL_FOLDER --vposer_ckpt VPOSER_FOLDER` \* for SMPL+H: `Shell python smplifyx/main.py --config cfg_files/fit_smplh.yaml --data_folder DATA_FOLDER --output_folder OUTPUT_FOLDER --visualize="True/False"--model_folder MODEL_FOLDER --vposer_ckpt VPOSER_FOLDER`

## Visualizing Results

To visualize the results produced by the method you can run the following script:

```
1 python smplifyx/render_results.py --mesh_fns OUTPUT_MESH_FOLDER
```

where *OUTPUT\_MESH\_FOLDER* is the folder that contains the resulting meshes.

## Dependencies

Follow the installation instructions for each of the following before using the fitting code.

1. PyTorch
2. SMPL-X
3. VPoser
4. Homogenous

## Optional Dependencies

1. PyTorch Mesh self-intersection for interpenetration penalty
  - Download the per-triangle part segmentation: *smplx\_parts\_segm.pkl* (ctrl/cmd +F for the filename)
2. Trimesh for loading triangular meshes
3. Pyrender for visualization

The code has been tested with Python 3.6, CUDA 10.0, CuDNN 7.3 and PyTorch 1.0 on Ubuntu 18.04.

---

## Citation

If you find this Model & Software useful in your research we would kindly ask you to cite:

```
1 @inproceedings{SMPL-X:2019,  
2   title = {Expressive Body Capture: 3D Hands, Face, and Body from a  
3     Single Image},  
4   author = {Pavlakos, Georgios and Choutas, Vasileios and Ghorbani,  
5     Nima and Bolkart, Timo and Osman, Ahmed A. A. and Tzionas,  
6     Dimitrios and Black, Michael J.},  
7   booktitle = {Proceedings IEEE Conf. on Computer Vision and Pattern  
8     Recognition (CVPR)},  
9   year = {2019}  
10 }
```

## Acknowledgments

### LBFGS with Strong Wolfe Line Search

The LBFGS optimizer with Strong Wolfe Line search is taken from this Pytorch pull request. Special thanks to Du Phan for implementing this. We will update the repository once the pull request is merged.

## Contact

The code of this repository was implemented by Vassilis Choutas and Georgios Pavlakos.

For questions, please contact [smplx@tue.mpg.de](mailto:smplx@tue.mpg.de).

For commercial licensing (and all related questions for business applications), please contact [ps-licensing@tue.mpg.de](mailto:ps-licensing@tue.mpg.de).