
Human Pose Estimation for Biomechanics

-

Wrapper instructions



Contents

1	Introduction	2
2	Run on Alvis	2
2.1	First Part of the Pipeline	2
2.2	Second Part of the Pipeline	3
3	Output	3
4	Improvement Suggestions	5

1 Introduction

Pipeline to perform pose estimation from single images. This provides Openpose 2D joints estimation, SMPLX and SMPL 3D pose estimation, OSSO skeleton estimation, performs rigid registration of the pelvic bone to get an appropriate position of the regressed 3D joints, and finally outputs a LS-Dyna keyfile to position the THUMS body model.

2 Run on Alvis

Check if you have access to the files in the following folder since this is where the containers used in this pipeline are stored:

/mimer/NOBACKUP/groups/snic2022-22-770/Azilis_workspace/singularity_image.

2.1 First Part of the Pipeline

Inputs:

```
workingFolder
├── images
│   └── example.jpg
├── keypoints (optional)
│   └── example_keypoints.json
├── bmi (optional)
│   └── example_bmi.json
```

- Format of the keypoints files: OpenPose keypoints output file
- Format of the bmi file (example):

```
{
  "height": 183.0,
  "weight": 130.0
}
```

On Alvis, cd to the wrapper directory then run:

```
sbatch main-part1.sh -f FOLDER_PATH -o true -b free
```

FOLDER_PATH should be the path to the working folder.

The following options are available:

1. -o (required): false if OpenPose keypoints are already available (in this case the openpose folder needs to be defined, only one person per image/keypoints case) or true if OpenPose keypoints are not available and thus running OpenPose is required.
2. -b (required): free if there are no constraints to be applied on the height and weight of the people of interest, fixed if the height and weight should be the same for all cases and personalized if there is height and weight information available for each single person. In this case, there should be only one person per image, and the bmi folder needs to be defined.

3. `-h` and `-w` (optional): if the option `-b fixed` is used, then these options should be defined as the fixed height (cm, float) and weight (kg, float).

2.2 Second Part of the Pipeline

Inputs:

```
workingFolder
├── Outputs of the first part
├── joints-blender
│   └── example_joints-blender.txt
```

Currently, the new joints have to be manually generated using Blender and the Meshcapade SMPL Add-on:

- In the Layout window, go to the Meshcapade add-on
- Generate an SMPLX avatar of the required gender (male, female, neutral)
- If needed, change the height and weight or directly the beta shape parameters (Editor > Data > Shape parameters from 00 to 09)
- For only one pose: Load the pose from the `.pkl` file saved in the results folder in part 1 of the pipeline, then from the Scripting window, run the `smplx_blender_single_file.py` script (don't forget to update the path to save the file)
- To iterate on the entire results folder: from the Scripting window, run the `smplx_blender_loop.py` script (don't forget to update the input folder path and the output folder path)
- Afterwards, all the joints files should be saved in or moved to the joints-blender folder in order to be able to run part 2

Then, run:

```
sbatch main-part2.sh -f FOLDER_PATH
```

3 Output

Example of the output structure for one input image with one person in it:

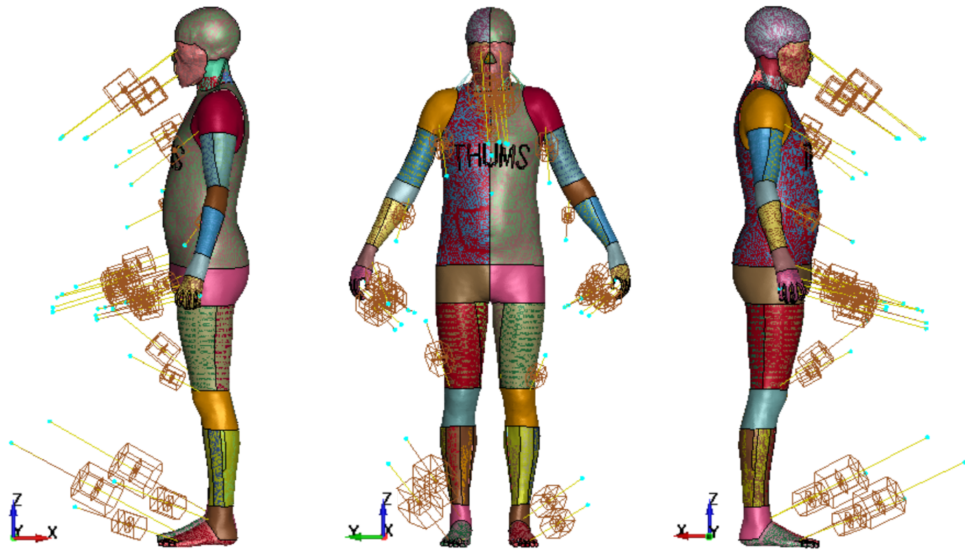
```
workingFolder
├── conf.yaml
├── images
│   └── example.jpg
├── images-rendered
│   └── example_rendered.png
├── joints
│   ├── example_joints.json
│   ├── joints-registered
│   └── example_joints-registered.json
```

```

├── example_joints-registered.ply
├── example_joints.ply
├── joints-blender
│   └── test_joints-blender.txt
├── joints-matlab
│   └── test_joints-registered.txt
├── keyfiles-ls-dyna
│   └── test_Positioning_cables.k
├── keypoints
│   └── example_keypoints.json
├── meshes
│   └── example
│       └── 000.obj
├── meshes-smpl
│   ├── example.obj
│   └── example.pkl
├── meshes-smplx
│   └── example.obj
├── osso
│   ├── example_skel_posed.ply
│   ├── registered
│   │   ├── example_skel_posed_registered.ply
│   │   └── example_skel_posed_rotated.ply
│   └── tmp
│       ├── example_skel_lying.pkl
│       ├── example_skel_lying.ply
│       ├── example_star_lying.ply
│       ├── example_star_posed.pkl
│       └── example_star_posed.ply
├── registration
│   ├── example-registration-matrix.txt
│   ├── pelvis-points
│   │   └── example_joints.ply
├── results
│   └── example
│       └── 000.pkl

```

Example of what the result looks like in LS-Dyna (before simulation to change the pose of the body model):



4 Improvement Suggestions

- Incorporate the Blender Add-on step directly into the pipeline to streamline the process and reduce manual intervention.