

---

# HumanNeRF: Free-viewpoint Rendering of Moving People from Monocular Video (CVPR 2022)

[Project Page](#) | [Paper](#) | [Video](#)

This is an official implementation. The codebase is implemented using PyTorch and tested on Ubuntu 20.04.4 LTS.

## Prerequisite

### Configure environment

Install Miniconda (recommended) or Anaconda.

Create and activate a virtual environment.

```
1 conda create --name humannerf python=3.7
2 conda activate humannerf
```

Install the required packages.

```
1 pip install -r requirements.txt
```

### Download SMPL model

Download the gender neutral SMPL model from [here](#), and unpack **mpips\_smplify\_public\_v2.zip**.

Copy the smpl model.

```
1 SMPL_DIR=/path/to/smpl
2 MODEL_DIR=$SMPL_DIR/smplify_public/code/models
3 cp $MODEL_DIR/basicModel_neutral_lbs_10_207_0_v1.0.0.pkl third_parties/
   smpl/models
```

Follow [this page](#) to remove Chumpy objects from the SMPL model.

### Run on ZJU-Mocap Dataset

Below we take the subject 387 as a running example.

---

## Prepare a dataset

First, download ZJU-Mocap dataset from [here](#).

Second, modify the yaml file of subject 387 at `tools/prepare_zju_mocap/387.yaml`. In particular, `zju_mocap_path` should be the directory path of the ZJU-Mocap dataset.

```
1 dataset:
2     zju_mocap_path: /path/to/zju_mocap
3     subject: '387'
4     sex: 'neutral'
5
6 ...
```

Finally, run the data preprocessing script.

```
1 cd tools/prepare_zju_mocap
2 python prepare_dataset.py --cfg 387.yaml
3 cd ../../
```

## Train/Download models

Now you can either download a pre-trained model by running the script.

```
1 ./scripts/download_model.sh 387
```

or train a model by yourself. We used 4 GPUs (NVIDIA RTX 2080 Ti) to train a model.

```
1 python train.py --cfg configs/human_nerf/zju_mocap/387/adventure.yaml
```

For sanity check, we provide a configuration that supports training on a single GPU (NVIDIA RTX 2080 Ti). Notice the performance is not guranteed for this configuration.

```
1 python train.py --cfg configs/human_nerf/zju_mocap/387/single_gpu.yaml
```

## Render output

Render the frame input (i.e., observed motion sequence).

```
1 python run.py \
2     --type movement \
3     --cfg configs/human_nerf/zju_mocap/387/adventure.yaml
```

Run free-viewpoint rendering on a particular frame (e.g., frame 128).

---

```
1 python run.py \  
2     --type freeview \  
3     --cfg configs/human_nerf/zju_mocap/387/adventure.yaml \  
4     freeview.frame_idx 128
```

Render the learned canonical appearance (T-pose).

```
1 python run.py \  
2     --type tpose \  
3     --cfg configs/human_nerf/zju_mocap/387/adventure.yaml
```

In addition, you can find the rendering scripts in `scripts/zju_mocap`.

## Run on a Custom Monocular Video

To get the best result, we recommend a video clip that meets these requirements:

- The clip has less than 600 frames (~20 seconds).
- The human subject shows most of body regions (e.g., front and back view of the body) in the clip.

## Prepare a dataset

To train on a monocular video, prepare your video data in `dataset/wild/monocular` with the following structure:

```
1 monocular |  
2   images | L  
3     ${item_id}.png |  
4   masks | L  
5     ${item_id}.png |  
6   metadata.json
```

We use `item_id` to match a video frame with its subject mask and metadata. An `item_id` is typically some alphanumeric string such as 000128.

**images** A collection of video frames, stored as PNG files.

**masks** A collection of subject segmentation masks, stored as PNG files.

---

**metadata.json** This json file contains metadata for video frames, including:

- human body pose (SMPL poses and betas coefficients)
- camera pose (camera intrinsic and extrinsic matrices). We follow OpenCV camera coordinate system and use pinhole camera model.

You can run SMPL-based human pose detectors (e.g., SPIN, VIBE, or ROMP) on a monocular video to get body poses as well as camera poses.

```
1 {
2   // Replace the string item_id with your file name of video frame.
3   "item_id": {
4     // A (72,) array: SMPL coefficients controlling body pose.
5     "poses": [
6       -3.1341, ..., 1.2532
7     ],
8     // A (10,) array: SMPL coefficients controlling body shape.
9     "betas": [
10      0.33019, ..., 1.0386
11    ],
12    // A 3x3 camera intrinsic matrix.
13    "cam_intrinsics": [
14      [23043.9, 0.0, 940.19],
15      [0.0, 23043.9, 539.23],
16      [0.0, 0.0, 1.0]
17    ],
18    // A 4x4 camera extrinsic matrix.
19    "cam_extrinsics": [
20      [1.0, 0.0, 0.0, -0.005],
21      [0.0, 1.0, 0.0, 0.2218],
22      [0.0, 0.0, 1.0, 47.504],
23      [0.0, 0.0, 0.0, 1.0],
24    ],
25  }
26  ...
27  // Iterate every video frame.
28  "item_id": {
29    ...
30  }
31 }
32 }
```

Once the dataset is properly created, run the script to complete dataset preparation.

```
1 cd tools/prepare_wild
2 python prepare_dataset.py --cfg wild.yaml
3 cd ../../
```

---

## Train a model

Now we are ready to launch a training. By default, we used 4 GPUs (NVIDIA RTX 2080 Ti) to train a model.

```
1 python train.py --cfg configs/human_nerf/wild/monocular/adventure.yaml
```

For sanity check, we provide a single-GPU (NVIDIA RTX 2080 Ti) training config. Note the performance is not guaranteed for this configuration.

```
1 python train.py --cfg configs/human_nerf/wild/monocular/single_gpu.yaml
```

## Render output

Render the frame input (i.e., observed motion sequence).

```
1 python run.py \  
2     --type movement \  
3     --cfg configs/human_nerf/wild/monocular/adventure.yaml
```

Run free-viewpoint rendering on a particular frame (e.g., frame 128).

```
1 python run.py \  
2     --type freeview \  
3     --cfg configs/human_nerf/wild/monocular/adventure.yaml \  
4     freeview.frame_idx 128
```

Render the learned canonical appearance (T-pose).

```
1 python run.py \  
2     --type tpose \  
3     --cfg configs/human_nerf/wild/monocular/adventure.yaml
```

In addition, you can find the rendering scripts in `scripts/wild`.

## Acknowledgement

The implementation took reference from NeRF-PyTorch, Neural Body, Neural Volume, LPIPS, and YACS. We thank the authors for their generosity to release code.

## Citation

If you find our work useful, please consider citing:

---

```
1 @InProceedings{weng_humannerf_2022_cvpr,
2   title      = {Human{N}e{RF}: Free-Viewpoint Rendering of Moving
3   People From Monocular Video},
4   author     = {Weng, Chung-Yi and
5   Curless, Brian and
6   Srinivasan, Pratul P. and
7   Barron, Jonathan T. and
8   Kemelmacher-Shlizerman, Ira},
9   booktitle  = {Proceedings of the IEEE/CVF Conference on Computer
10  Vision and Pattern Recognition (CVPR)},
11  month      = {June},
12  year       = {2022},
13  pages      = {16210-16220}
14 }
```