

**This Guide Assumes You Have EasyMocap, OpenPose, and the required models already downloaded**

- <https://github.com/zju3dv/EasyMocap/blob/master/doc/installation.md>

**Capturing data:**

1. Set up cameras for best results
  - a. Equidistant from the center of set
  - b. Framed such that actor will be in frame on as many cameras as possible
  - c. Use high shutter to prevent motion blur
  - d. Avoid having objects such as wires or other people in frame of cameras
2. Calibration
  - a. Recording
    - i. Intrinsic
      1. Stand at the same distance from your camera that you plan to record mocap
      2. Rotate and move the chessboard in front of camera with a clear view
        - a. Be sure to get different angles of chessboard by
        - b. These clips don't need to be synchronized
        - c. NOTES:
          - i. Maximize your time and variety in rotations as the longer the intrinsic video the longer the calibration will take
        - d. Be sure to bring the chessboard the edges of the frame or your footage will be distorted
        - e. Note:
          - i. I usually start chessboard in center of frame and move it to the left edge then to the right edge, then rotate it towards me so it's about 20 degrees from flat
    - ii. Extrinsic
      1. Set the chessboard in the center of your mocap space and record a short video on all the cameras
  - b. Generating Calibration Files
    - i. Intrinsic
      1. In a folder, create a new folder called videos
        - a. Label these videos 1, 2, 3, etc
        - b. Let **`${intri}`** = **this**
      2. In command line
        - a. `python3 ${location of EasyMocap files}/scripts/preprocess/extract_video.py ${intri} --no2d`
        - b. `python3 ${location of EasyMocap files}/apps/calibration/detect_chessboard.py ${intri} --out`

**`${location referenced in 1}/output/calibration --pattern y,x --grid 0.1`**

- i. Y = number of tiles in y direction - 1
- ii. X = number of tiles in x direction - 1

3. fine tuning

- a. **`python3 ${location of EasyMocap}/apps/annotation/annot_calib.py ${intri} --mode chessboard --pattern y,x --annot chessboard`**
  - i. Y = number of tiles in y direction - 1
  - ii. X = number of tiles in x direction - 1
- b. This will open a window with a zigzag on the chessboard. Go through the frames with A and D.
  - i. If there is a frame that isn't matching the others
    - 1. Press space until the red circle is on the point you wish to change
    - 2. Then click where you want the point to be and press space
  - ii. Once all frames are correctly set press q and you do this for the next camera

ii. Extrinsic

- 1. In a folder, create a new folder called videos
  - a. Label these videos 1, 2, 3, etc
  - b. Let **`${extri} = this`**
- 2. In command line
  - a. **`python3 ${location of EasyMocap files}/scripts/preprocess/extract_video.py {location referenced in 1} --no2d`**
  - b. **`python3 ${location of EasyMocap files}/apps/calibration/detect_chessboard.py ${extri} --out ${extri}/output/calibration --pattern y,x --grid 0.1`**
    - i. Y = number of tiles in y direction - 1
    - ii. X = number of tiles in x direction - 1

3. finetuning

- a. **`python3 ${location of EasyMocap}/apps/annotation/annot_calib.py ${extri} --mode chessboard --pattern y,x --annot chessboard`**
  - i. Y = number of tiles in y direction - 1
  - ii. X = number of tiles in x direction - 1
- b. You can fine tune this the same way you finetuned intrinsic
  - i. But since each shot is stationary you only need 1 frame per camera to be accurate and you can delete the rest.
    - 1. So you just need to fix 1 frame per camera and delete the other json files in chessboard

**c. Note:**

- i. A major difference in extrinsic is that they are synced.
  - ii. So you need to make sure the zigzag is in the same relative position on each camera
    1. This is how the cameras figure out where they are
- iii. Generating files for mocap
1. Intrinsic
    - a. python3 **\${location of EasyMocap}/apps/calibration/calib\_intri.py \${intri}**
      - i. This will take a lot of time
  2. Extrinsic
    - a. python3 **\${location of EasyMocap}/apps/calibration/calib\_extri.py \${extri} --intri \${intri}/output/intri.yml**
  3. python3 **\${location of EasyMocap}/apps/calibration/check\_calib.py \${extri} --out \${extri} --show**
    - a. This will show an image with a box drawn on
    - b. Press D to go to the next camera and verify the box is in the same relative location

**c. Generating Mocap**

- i. Create a project folder
  1. In it create a videos folder containing the **SYNCED** videos you want to mocap
    - a. Label these videos 1, 2, 3, etc
  2. Let **\${data}** = project folder
  3. Copy the extri.yml and intri.yml files from **\${extri}** into **\${data}**
- ii. In command line
  1. python3 **\${location of EasyMocap}/scripts/preprocess/extract\_video.py \$data --openpose \${location of openpose data(inside folder)} --handface --gtbbox**
  2. Get nohands or face
    - a. python3 **\${location of EasyMocap}/apps/demo/mv1p.py \${data} --out \${data}/output/smpl --vis\_det --vis\_repro --undis --sub\_vis 1 2 3 etc --vis\_smpl**
  3. Hands and face
    - a. python3 **\${location of EasyMocap}/apps/demo/mv1p.py \${data} --out \${data}/output/smplx --vis\_det --vis\_repro --undis --sub\_vis 1 2 3 etc --body bodyhandface --model smplx --gender male --vis\_smp**

**iii. Tips:**

1. If your 3d character in the output/smpl images seems to be too small
  - a. Manually go into the extri.yml and divide the numbers in the data array of T\_1, T\_2, ...
    - i. If a certain camera has a much smaller or much larger character just change the data values in the correlating T\_{**camera number**}
2. If using selfie camera of cellphone be sure to flip the footage, as it may be reversed, or you will get confusing results
3. If you notice that your footage is distorted in the check mocap step try adding the -tryfocal
4. If you are getting an index error its probably being caused by poor detection
  - a. So you can reshoot it or you can use -end [insert number] at a frame the precedes the poor detection
    - i. You can find detection in **`\${data}`**/output/detect
5. AssertionError
  - a. Your missing files somewhere / they are in the wrong place
  - b. You already generated an output and are doing so again
    - i. If you want to regenerate Mocap either move or delete previous output