

---

# Fast and Robust Multi-Person 3D Pose Estimation from Multiple Views

Fast and Robust Multi-Person 3D Pose Estimation from Multiple Views

Junting Dong, Wen Jiang, Qixing Huang, Hujun Bao, Xiaowei Zhou

CVPR 2019, T-PAMI 2021 Project Page

Any questions or discussions are welcomed!

## Installation

- Set up python environment

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

- Compile the `backend/tf_cpn/lib` and `backend/light_head_rcnn/lib`

```
1 cd mvpose/backend/tf_cpn/lib/  
2 make  
3 cd ../lib_kernel/lib_nms  
4 bash compile.sh
```

```
1 cd mvpose/backend/light_head_rcnn/lib/  
2 bash make.sh
```

Since they use py-faster-rcnn as backbone. Many people using faster-rcnn meet with some problems when compiling those components. Suggestions on google can be helpful.

- Compile the pictorial function for acceleration

```
1 cd mvpose/src/m_lib/  
2 python setup.py build_ext --inplace
```

## Prepare models and datasets

- **Prepare models:** Please put light-head-rcnn models to `backend/light_head_rcnn/output/model_dump`, `backend/tf_cpn/log/model_dump` to `backend/tf_cpn/log/model_dump`, and CamStyle model trained by myself to `backend/CamStyle/logs`
- **Prepare the datasets:** Put datasets such as Shelf and CampusSeq1 to `./datasets/`. Download Campus and Shelf datasets. Then, put datasets such as Shelf and CampusSeq1 to `datasets/`
- **Generate the camera parameters:** Since each dataset uses different way to obtain the camera parameters, we show an example to deal with the Campus dataset:

- 
- Add following code to `.datasets/CampusSeq1/Calibration/producePmat.m`

```
1 K = cell(1,3);
2 K{1} = K1; K{2} = K2; K{3} = K3;
3 m_RT = cell(1,3);
4 m_RT{1} = RT1; m_RT{2} = RT2; m_RT{3} = RT3;
5 save('intrinsic.mat','K');
6 save('m_RT.mat', 'm_RT');
7 save('P.mat', 'P');
8 save('prjectionMat','P');
```

- generate the `camera_parameter.pickle`

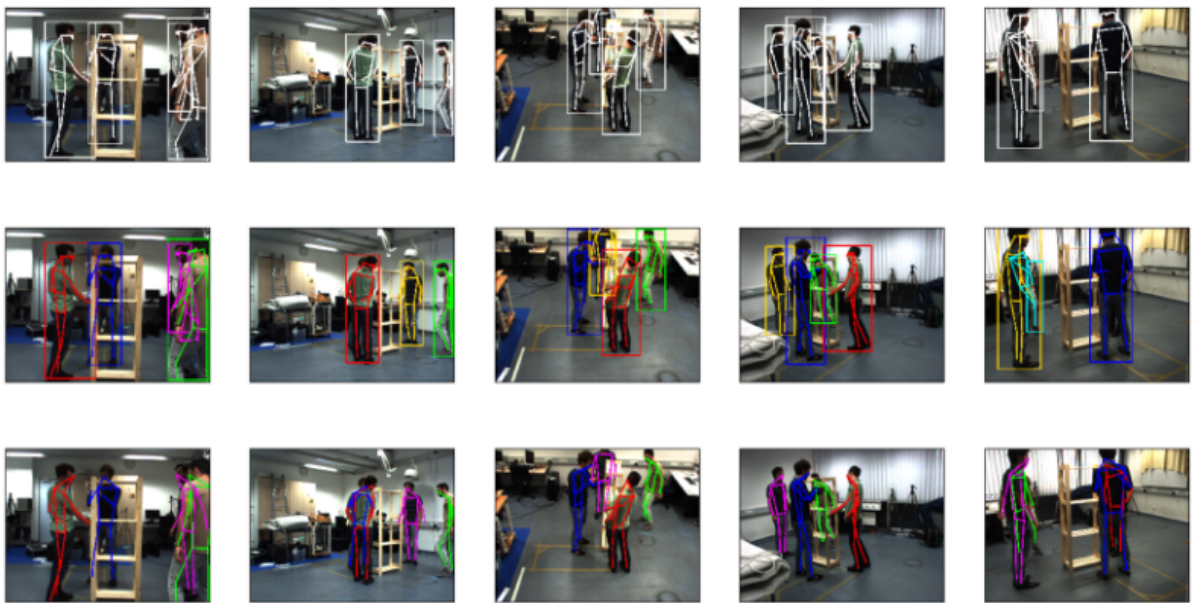
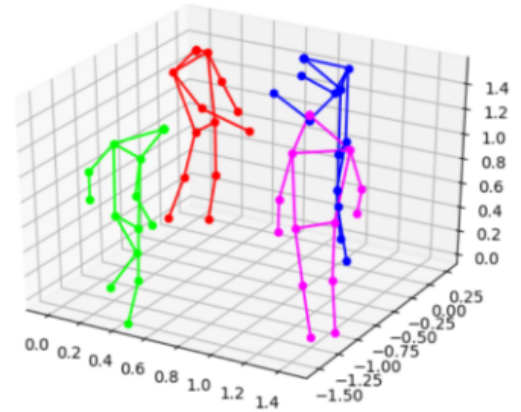
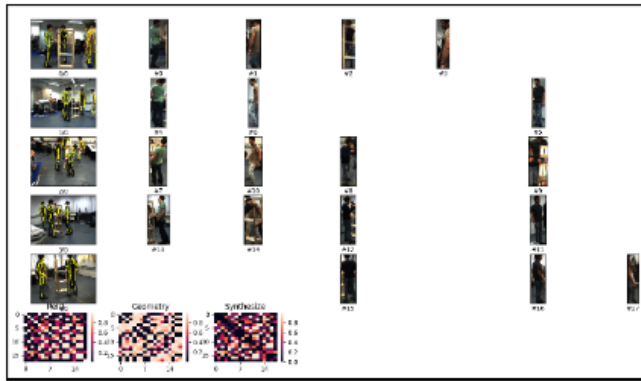
```
1 python ./src/tools/mat2pickle.py /parameter/dir ./datasets/
   CampusSeq1
```

Here, we also provide the `camera_parameter.pickle` of Campus and Shelf. You can generate the `.pickle` file for your datasets using the same way. ## Demo and Evaluate

### Run the demo

```
1 python ./src/m_utils/demo.py -d Campus
2 python ./src/m_utils/demo.py -d Shelf
```

If all the configuration is OK, you may see the visualization of following items.



## Evaluate on the Campus/Shelf datasets

```
1 python ./src/m_utils/evaluate.py -d Campus
2 python ./src/m_utils/evaluate.py -d Shelf
```

As long as the progress bar finished, you may see a beautified table of evaluation result and a csv file for the evaluation result will be save in `./result` directory.

## Accelerate the evaluation

Since the 2D pose estimator (CPN) is a little slow, we can save the predicted 2D poses and heatmaps and then start with these saved files.

---

1. produce the files

```
1 python src/tools/preprocess.py -d Campus -dump_dir ./datasets/  
   Campus_processed  
2 python src/tools/preprocess.py -d Shelf -dump_dir ./datasets/  
   Shelf_processed
```

2. evaluate with saved 2D poses and heatmaps

```
1 python ./src/m_utils/evaluate.py -d Campus -dumped ./datasets/  
   Campus_processed  
2 python ./src/m_utils/evaluate.py -d Shelf -dumped ./datasets/  
   Shelf_processed
```

Note: for the sake of convenience, we do not optimize on the size of dumped file. Therefore, the size of `Campus_processed` is around 4.0G and the size of `Shelf_processed` is around 234G. Please make sure your disk have 200+G free space. Any pull request to solve this issues will be welcomed.

## Citation

If you find this code useful for your research, please use the following BibTeX entry.

```
1 @inproceedings{dong2019fast,  
2   title={Fast and Robust Multi-Person 3D Pose Estimation from Multiple  
   Views},  
3   author={Dong, Junting and Jiang, Wen and Huang, Qixing and Bao, Hujun  
   and Zhou, Xiaowei},  
4   journal={CVPR},  
5   year={2019}  
6 }  
7  
8 @article{dong2021fastpami,  
9   title={Fast and Robust Multi-Person 3D Pose Estimation and Tracking  
   from Multiple Views},  
10  author={Dong, Junting and Fang, Qi and Jiang, Wen and Yang, Yurou and  
   Bao, Hujun and Zhou, Xiaowei},  
11  booktitle={T-PAMI},  
12  year={2021}  
13 }
```

## Acknowledgements

This code uses these code (Light head rcnn, Cascaded Pyramid Network, CamStyle) as backbone. We gratefully appreciate the impact it had on our work. If you use our code, please consider citing the original paper as well.

---

## Copyright

This work is affiliated with ZJU-SenseTime Joint Lab of 3D Vision, and its intellectual property belongs to SenseTime Group Ltd.

```
1 Copyright SenseTime. All Rights Reserved.
2
3 Licensed under the Apache License, Version 2.0 (the "License");
4 you may not use this file except in compliance with the License.
5 You may obtain a copy of the License at
6
7     http://www.apache.org/licenses/LICENSE-2.0
8
9 Unless required by applicable law or agreed to in writing, software
10 distributed under the License is distributed on an "AS IS" BASIS,
11 WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied
12 .
13 See the License for the specific language governing permissions and
14 limitations under the License.
```