A comparison of human skeleton extractors for real-time human-robot interaction

Wanchen Li, Robin Passama, Vincent Bonnet, Andrea Cherubini

A comparison of human skeleton extractors for real-time human-robot interaction

Wanchen LI Robin Passama Vincent Bonnet Andrea Cherubini

***Abstract*—Modern industrial manufacturing procedures gradually integrate physical Human-Robot interaction (pHRI) scenarios. This requires robots to understand human intentions for effective and safe cooperation. Vision is the most commonly used sensor modality for robots to perceive human behavior. In this paper, a comparison of existing vision-based human skeleton extraction frameworks is made, to provide guidance for the design of human-robot interaction applications. A dataset consisting of consecutive images that records 14 actions conducted by different users acquired by a kinect camera is used for human skeleton extraction. The work justifies our choice of skeleton extractors according to pHRI constraints.**

***Index Terms*—human-robot interaction, activity recognition, 3D skeleton detection, computer vision**

I. Introduction

A. Motivation for human activity recognition using skeleton data

Modern manufacturing processes are integrating more and more human-robot cooperation scenarios, where robot and humans share the same workspace. To successfully perform a cooperation task, robots need to embed intelligence to be able to understand the intention of humans, by recognizing human activity. This is first of all for safety considerations: since the robot is located in proximity to the human, the movement of the robot should be guaranteed to be harmless to human. For this reason, the robot needs to dispose information on human body posture. Moreover, since robots and human are sharing the same workspace, the two can perceive each other's behavior and even physically interact directly or indirectly. The capability of understanding human activity enables robots to collaborate with humans in an intuitive and efficient manner. Apart from improving productivity, human robot cooperation should also enhance human comfort and prevent potential musculoskeletal disorders. With clever definition of a taxonomy of activities, activity recognition provides a basis for online ergonomic evaluation of human postures [15].

Currently, one of the most popular data sources for human activity recognition is vision based data, since visual modality has been continuously studied and significant advances in both computer science and hardware have been made in recent years. With recent release of cost-effective depth sensors, the problem of loss of information in projection from 3d to 2d can be recovered. Depth cameras give easy access to 3d human body data at a high resolution frame, therefore, this data acquisition mode is close to human-being perception. The low-cost and ease of installation of depth camera, compared to motion capture systems such as xsens, vicon makes it more friendly for use cases in the complex industrial human-robot cooperation context. The depth camera is used in a non invasive manner, which does not require workers to carry addition loads. Therefore, we pay special attention to human activity recognition technology based on 3d visual perception.

In [1], the authors compared human activity recognition from different data sources of range sensors, i.e. single devices that can capture 3D data from one point of view, such as depth camera. The comparison was made between 3D silhouettes, skeletal data, spatio-temporal features, 3D occupancy features extracted from sensors, and 3D optical flow data. The review showed that using 3D silhouettes for activity recognition can be only limited for atomic actions recognition and occlusion can gravely degrade the silhouettes extraction. This is not suitable for human-robot scenarios including complex actions and interactions with objects. Using 3d optical flow data is computationally costly and not suitable for real time application. Using skeletal data for feature building is invariant to camera location and subject appearance, and it is better at modeling finer activities. Indeed, skeleton-based recognition exploits the position and orientation changes of human joints between frames. Since the joints and the segments structure can be translated into a kinematic model of human body, the kinematic properties of different body parts can provide abundant features for activity recognition. Of course, skeletal data can't provide information on external objects involved in the human activities, and any incompleteness of skeleton data will largely degrade the recognition.

B. Previous work on 3d skeletal data acquisition from depth camera

Skeleton data is an auxiliary output of raw visual data for human activity recognition. It generates a vast literature and tons of astonishing outbreaks of technology improvements in recent years. Skeletal data can be acquired from pure RGB data or in combination with depth information.

Initial skeleton extraction methods merged from the works on body part extraction, identification and detection using classifiers on monocular images [3] [5]. In [11], the technology of joint position identification using a deep randomized decision forest based on body part proposal from depth image

W.Li, R.Passama, A.Cherubini are with LIRMM, University of Montpellier, CNRS, Interactive Digital Humans group, 161 rue Ada, 34095, Montpellier, France. firstname.lastname@lirmm.fr;

V.Bonnet is with LAAS-CNRS, INSA-Toulouse, 7 Av. du Colonel Roche, 31400 Toulouse, France. firstname.lastname@laas.fr

was proposed, this became the core technology embeded in Microsoft kinect Xbox. Another notable work was using joint regressors based on random forests algorithm for estimation 2d skeleton positions [7]. Afterwards, with the advancement on machine learning technology, skeleton extraction by learning showed great results [13] [14]. Another side effect of research on machine learning is the explosive growth of data volume. Many free-access large datasets have been created for launching different algorithmic challenge so as to promote technical advancement in machine learing. Among them, for the topic of human pose estimation (or skeleton recognition), famous datasets are MPII Human Pose [2], OCHuman [17], Human3.6M [10]. The MPII dataset is composed of images taken from online videos covering hundreds of human activities, with person's 16 joints positions manually annotated. The Human3.6M is a motion capture dataset, images are acquired from motion capture systems with accurate 3D joint positions recording. The OCHuman is a dataset that focuses on occluded human images, which makes it the most challenging dataset for human detection. Among these 3, the MPII dataset is the most coherent to our scope of discussion. We browsed the relevant papers on human skeleton recognition developed based on this dataset. We screened several frameworks that are most likely to be suitable for real-time human-robot collaboration applications for comparison. Our selection criteria are as follows:

- The proposed framework for skeleton extraction has open access code source. In addition, the code source should provide simple APIs for direct downstream exploitation. It should be portable for arbitrary image or video inference. The input image or video for inference does not carry additional key-point cues or multiple view-angle cues.
- The proposed framework should be able to extract skeleton from images at a considerable speed. Because for the pHRI scenario, the human activity recognition task must be fast in order to timely control the robot. In our specific pHRI scenario, we tend to use skeleton extraction libraries to obtain Cartesian coordinates of joint positions of human body, then apply inverse kinematic analysis on the acquired skeleton information to obtain joint angles, finally use joint angles to recognize human activity [12]. In this complete pipeline, to assure its real time compatibility, the skeleton extraction algorithm should run at about 10 Hz. The algorithm should at least be capable of providing skeleton recognition information of limbs and torso, to assure a successful inverse kinematic analysis on human body model.
- Our pipeline is developed in a C++ implementation framework, ideally the skeleton extraction library should allow easy interfacing with the current framework.

In fact, the first restriction has already excluded a large part of skeleton extraction frameworks. In the following section, we will discuss several well developed frameworks that are compatible with the first restriction. We discuss their skeleton extraction principles and their execution performances. For this, we test each library on a video recorded during an experimentation for SOPIHA project at University of Montpellier. In this video, a person used a brush to deburr a gear in cooperation with BAZAR robot, which simulates an industrial activity in a pHRI scenario. There is the presence of different objects, a single person and 2 robotic arms in the video. The test was performed on a computer equipped with 1 GPU GeForce GTX 1060 and 1 CPU Intel(R) Core(TM) i7-6700HQ CPU @ 2.60GHz. Our selected frameworks for skeleton extraction are Detectron2, Mediapipe, Yolov7, Alphapose and Openpose. All of these frameworks rely on deep learning method for skeleton prediction.

II. Comparison of human skeleton extraction frameworks

A. Detectron2

Detectron2 is a library that provides human detection and segmentation algorithms for computer vision inputs such as image or video. It is the next generation of Detectron library, developed by Facebook AI Research (FAIR). The library is implemented in Python upon Pytorch engine, it embeds various variant algorithm of Mask R-CNN model developed also by FAIR [9]. Its advantages with respect to prior technology Faster-RCNN, is the high-quality on instance segmentation, by adding a branch in the network for providing object mask with the existing branch for bounding box recognition. The skeleton detection is an extended functionality. The framework regards each key-point of human pose as a one-hot mask, the prediction of key-points on human skeleton simply becomes the prediction of K object masks on the human body zone, one for each key-point located on the joint position. These keypoint prediction is made without knowledge on human body modeling structure. As such, this is a one shot detection method applicable for multi-person cases. The more the presence of human in the visual input, and the more key-points to be detected, the longer the algorithm needs for calculation.

The given model in the library for key-point detection was trained on COCO `trainval35k`. The framework offers possibility for customized training. For keypoint detection, the output of the framework is the bouding box position of detected person, his joint positions, the segmentation mask and the confidence level of the prediction. We run the framework on our customised video input, the prediction runs at around 3.57 fps on the tested computer. We observe strong oscillation on the predicted joint positions between frames. Keypoint prediction is missing on occluded human body part. The result certainly fails to meet our needs for real time skeleton recognition in terms of computing speed and accuracy.

B. Mediapipe

Mediapipe is a library for body pose tracking in RGB video frames using a 33 points topology. The algorithm is based on BlazePose [4], it is a two-step encoder-decoder method. A detector firstly locates the region of interest within the frame, then a tracker locates the joint positions and tracks them in the

derived ROI in the upcoming frames. With this principle, the detection only work with the presence of person face because the person detector is built on the face detection proxy, upon assumption that the head of person should always be visible. In addition, the detection of the person does not happen on each frame in order to gain a rapid execution speed. These principles make it a light-weight detection library but only applicable for vision streaming input which contains a single person whose movement does not involve sudden changes between frames. The framework is based on Tensorflow engine, written in C++ with possible APIs in Python via Pybind library and in C++ via Bazel building.

Mediapipe is trained on closed-source dataset, it can provided 3d inference of joint positions, the z direction coordinates are inferred from the pairing of detected x, y joint positions with Ghum 3D model (Generative 3D Human Shape and Articulated Pose Models). The library achieves a processing speed on about 17 fps on our video input. Due to its inference principle, we observe that the detected key-point positions are largely biased from actual joint positions when the person is moving fast in the video. Keypoint detection is available on occluded body parts, but once again, the accuracy of these detected positions is questionable.

C. Yolov7

Yolov7 is an architecture capable of object detection, object segmentation and human skeleton detection, which differs it from previous YOLO models. The algorithm optimized the training cost, the re-parameterized method, dynamic label assignment strategy and model scaling techniques with respect to previous YOLO algorithms, this enables faster inference speed and higher detection accuracy [16]. It is a single stage detector implementing a heatmap free approach for keypoint detection. It can provides one-shot detection for multi-person. The framework is implemented in Python based on Pytorch which allows customized training and the provided model for keypoint detection was trained on COCO dataset, which contains annotations on body segment but not on hand, face and foot. The processing speed of this architecture on our video is 11.04 fps. The prediction results on joint positions are stable between frames. The algorithm can even output keypoint predictions on occluded body part.

D. Alphapose

Alphapose is an architecture proposed in 2018 and the development community remains actively updated today in 2023. The pose detection algorithm follows the top-down strategy. Compared to other top-down skeleton extraction algorithms, its difference is using a lower detection confidence for human detection, with this the algorithm can obtain more candidates for pose estimation. Pose estimation candidates are then eliminated by parametric pose non-maximum-suppression. For dealing with scaling problem across different body part keypoint estimation, the algorithm adopts symmetric integral keypoints regression method to localize keypoints in different scales. The algorithm also introduce a pose-aware identity embedding to conduct identity tracking duing pose estimation [8]. The framework is implemented in Python based on Pytorch, and different pretrained models by COCO, PoseTrack and Halpe-FullBody datasets are proposed for either torso or whole body pose estimation including face and hands. We tested different pretrained models in the Alphapose library on our video. Our obtained best performance in terms of execution speed is using Fast Pose DUC model for 17 keypoints estimation in COCO format at 9.74 fps, using Fase Pose model for 26 keypoints estimation in Halpe format at 8.01 fps, and ??? for 136 keypoints estimation. The pose estimation results oscillate between adjacent frames, and pose estimation result is missing on occluded body part.

E. Openpose

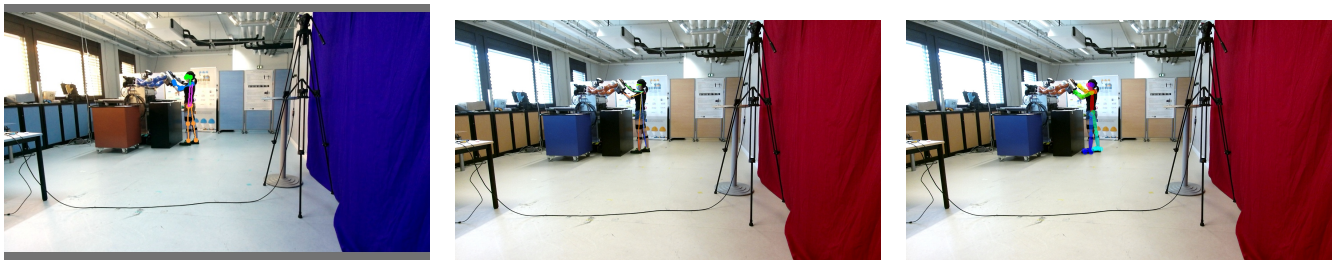
Openpose is a skeleton extraction library for multi-person detection of body, hand, foot and facial keypoints. It follows a buttom-up approach. The keypoint estimation on face and hand; the 3d triangulation for all detected keypoints are both relied on multi-camera system, for our use case with one single camera, the library can only offer 25 keypoints estimations in 2d on body and foot. For this, the library uses a feed-forward network to predict a confidence map of body part locations and a set of part affinity fields. Then these confidence maps and part affinity fields are parsed by greedy inference in the global context of detection to output 2d keypoints [6]. This core model for body-foot keypoint detection is trained on COCO and MPII datasets, plus a small subset of foot instances out of COCO datasets with 6 additional annotated keypoints.

The library is implemented based on caffe deep learning engine, in C++ language. We test its core functionality of body-foot keypoint preciction on our custom video, the processing speed has reached 8.7 fps for outputting the result at default resolution 1x368. At this resolution level, the output keypoints have slight ocsilations between frames but the global skeleton prediction result remains convincing. We can further speed up the processing by tuning output resolution to a lower value with sacrifice of detection accuracy. We found that when the output resolution has been lower to 1x320, the skeleton output remains convincing without strong ocsilation between frames, and the processing speed at this resolution is about 9.91 fps.

III. Conclusions

Apart from above 5 libraries, another open-source library with complete implementation and interfaces for arbitrary images and videos skeleton extraction is Mmpose library. This library has integrated a wide spectrum of skeleton extraction algorithms. We have also tested this library on our video but the processing speed was about 2.155 fps. Due to its inefficiency, we didn't include it in the library comparison. ArtTrack and LightTrack libraries are also excluded for the same reason, MeTRAbs is excluded because of its close dependency on a 3d display rendering software.

We summary our comparison between abovementioned libraries in the below table I, we quantify their corresponding processing speed. Our running test is carried out on an old



(a) Yolo output. (b) Alphapose output. (c) Openpose output.

Fig. 1: Skeletion estimation output from YOLO, Alphapose, Openpose on a same image.

TABLE I

Skeleton extractors	Tracking	Multi-person detection	Foot keypoints	Hand keypoints	Facial keypoints	Easy C++ interfacing	Framerate
Detectron2	×	✓	×	×	ears,eyes,nose	×	3.57 fps
Mediapipe	✓	×	✓	✓	ears,eyes, nose,mouth	×	17 fps
YOLOv7	×	✓	×	×	ears,eyes,nose	×	11.04 fps
Alphapose	✓	✓	×	×	ears,eyes,nose	×	9.74 fps
Openpose	×	✓	✓	×	ears,eyes,nose	✓	9.91 fps

model graphics card, if it is configured on a advanced graphics card, there is still room for processing speed improvement for each library.

For our project, the desired pipeline which implements the skeleton extraction framework is written in C++ language. Therefore, in this comparison we care about the easiness of interfacing the skeleton extraction library with c++ coding project. In the above 5 libraries, the most adapted one should be Openpose. However, this is a constraint unique for our project. Broadly speaking, for general pHRI applications, YOLOv7, Alphapose and Openpose libraries are all worth considering. To further compare these 3 libraries, we will focus on their skeleton extraction accuracy.

It is difficult to compare the accuracy from the visual feedback of the skeleton detection results of different libraries(see Fig1), they presumably all give credible skeleton predictions. We need a quantitative accuracy comparison of these skeleton extraction libraries, this is done in the next section between YOLOv7, Alphapose and Openpose libraries. Detectron2 library is excluded because of its inefficiency, and we also excluded Mediapipe because its keypoint detection is limited in single person use case with no significant movement, while these constraints can be easily violated in industrial application.

ACKNOWLEDGMENT

This research was carried out in the context of the SOPHIA project, which received funding from the EU Horizon 2020 research and innovation program under Grant Agreement No. 871237.

REFERENCES

[1] AGGARWAL, J., AND XIA, L. Human activity recognition from 3d data: A review. *Pattern Recognition Letters 48* (2014), 70–80. Celebrating the life and work of Maria Petrou.

[2] ANDRILUKA, M., PISHCHULIN, L., GEHLER, P., AND SCHIELE, B. 2d human pose estimation: New benchmark and state of the art analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2014).

[3] ANDRILUKA, M., ROTH, S., AND SCHIELE, B. Pictorial structures revisited: People detection and articulated pose estimation. In *2009 IEEE Conference on Computer Vision and Pattern Recognition* (2009), pp. 1014–1021.

[4] BAZAREVSKY, V., GRISHCHENKO, I., RAVEENDRAN, K., ZHU, T., ZHANG, F., AND GRUNDMANN, M. BlazePose: On-device Real-time Body Pose tracking. *arXiv e-prints* (June 2020), arXiv:2006.10204.

[5] BOURDEV, L., AND MALIK, J. Poselets: Body part detectors trained using 3d human pose annotations. In *2009 IEEE 12th International Conference on Computer Vision* (2009), pp. 1365–1372.

[6] CAO, Z., SIMON, T., WEI, S.-E., AND SHEIKH, Y. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (July 2017).

[7] DANTONE, M., GALL, J., LEISTNER, C., AND VAN GOOL, L. Human pose estimation using body parts dependent joint regressors. In *2013 IEEE Conference on Computer Vision and Pattern Recognition* (2013), pp. 3041–3048.

[8] FANG, H.-S., LI, J., TANG, H., XU, C., ZHU, H., XIU, Y., LI, Y.-L., AND LU, C. Alphapose: Whole-body regional multi-person pose estimation and tracking in real-time. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022), 1–17.

[9] HE, K., GKIOXARI, G., DOLLAR, P., AND GIRSHICK, R. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (Oct 2017).

[10] IONESCU, C., PAPAVA, D., OLARU, V., AND SMINCHISESCU, C. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence 36*, 7 (2014), 1325–1339.

[11] SHOTTON, J., FITZGIBBON, A., COOK, M., SHARP, T., FINOCCHIO, M., MOORE, R., KIPMAN, A., AND BLAKE, A. Real-time human pose recognition in parts from single depth images. In *CVPR 2011* (2011), pp. 1297–1304.

[12] SINGH, A. K., ADJEL, M., BONNET, V., PASSAMA, R., AND CHERUBINI, A. A framework for recognizing industrial actions via joint angles.

In *2022 IEEE-RAS 21st International Conference on Humanoid Robots (Humanoids)* (2022), pp. 210–216.

[13] TOMPSON, J. J., JAIN, A., LECUN, Y., AND BREGLER, C. Joint training of a convolutional network and a graphical model for human pose estimation. In *Advances in Neural Information Processing Systems* (2014), Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, Eds., vol. 27, Curran Associates, Inc.

[14] TOSHEV, A., AND SZEGEDY, C. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2014).

[15] VIANELLO, L., GOMES, W., STULP, F., AUBRY, A., MAURICE, P., AND IVALDI, S. Latent ergonomics maps: Real-time visualization of estimated ergonomics of human movements. *Sensors 22*, 11 (2022).

[16] WANG, C.-Y., BOCHKOVSKIY, A., AND LIAO, H.-Y. M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv e-prints* (July 2022), arXiv:2207.02696.

[17] ZHANG, S.-H., LI, R., DONG, X., ROSIN, P., CAI, Z., HAN, X., YANG, D., HUANG, H., AND HU, S.-M. Pose2seg: Detection free human instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2019).