Master Thesis

Czech
Technical
University
in Prague

F3

Faculty of Electrical Engineering
Department of Cybernetics

Close proximity human keypoint detection

Jan Dočekal

Supervisor: Mgr. Matěj Hoffmann, Ph.D.
Supervisor–specialist: Ing. Jakub Rozlivek
Field of study: Open Informatics
May 2022

MASTER‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name:	**Dočekal Jan**	Personal ID number:	**478074**
Faculty / Institute:	**Faculty of Electrical Engineering**		
Department / Institute:	**Department of Computer Science**		
Study program:	**Open Informatics**		
Specialisation:	**Artificial Intelligence**		

II. Master’s thesis details

Master’s thesis title in English:

Close proximity human keypoint detection

Master’s thesis title in Czech:

Detekce částí lidského těla z velké blízkosti

Guidelines:

Monitoring the distance between robot and human body parts is essential for safe human-robot interaction. In close interaction, the robot often does not see the complete body of the person it is interacting with. The focus of this work is to localize the human body parts in an image from the robot’s view, with special attention to the upper body, arms, and hands.
1. Check the literature on human body keypoint detection from RGB and RGB-D inputs with outputs in 2D (image) or 3D.
2. Survey existing datasets with humans in close proximity to a camera - only certain body parts such as hands are visible. If needed, consider creating your own dataset.
3. Develop a method for close proximity human body keypoint detection in 2D (input image). If time permits, consider tracking - exploiting temporal correlations between frames.
4. Transform 2D keypoints to 3D with or without using depth provided by the RGB-D sensor. Experimentally validate the robustness and accuracy of the proposed method.
5. Demonstrate your method in a human-robot interaction scenario (iCub robot with Intel RealSense RGB-D camera on the head).

Bibliography / sources:

[1] Cao, Z., Hidalgo, G., Simon, T., Wei, S. E., & Sheikh, Y. (2019). OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields. IEEE Transactions on Pattern Analysis and Machine Intelligence, 43(1), 172-186.
[2] Cheng, Y., Yang, B., Wang, B., Yan, W., & Tan, R. T. (2019). Occlusion-aware networks for 3d human pose estimation in video. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 723-732).
[3] Fang, H. S., Xie, S., Tai, Y. W., & Lu, C. (2017). RMPE: Regional multi-person pose estimation. In Proceedings of the IEEE international conference on computer vision (pp. 2334-2343).
[4] Güler, R. A., Neverova, N., & Kokkinos, I. (2018). Densepose: Dense human pose estimation in the wild. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 7297-7306).
[5] Zhang, F., Bazarevsky, V., Vakunov, A., Tkachenka, A., Sung, G., Chang, C., & Grundmann, M. (2020). MediaPipe Hands: On-device Real-time Hand Tracking. ArXiv, abs/2006.10214.

Name and workplace of master’s thesis supervisor:

Mgr. Matěj Hoffmann, Ph.D. Vision for Robotics and Autonomous Systems FEE

Name and workplace of second master’s thesis supervisor or consultant:

Date of master’s thesis assignment: **21.01.2022** Deadline for master's thesis submission: ___________

Assignment valid until: **30.09.2023**

Mgr. Matěj Hoffmann, Ph.D. Supervisor’s signature	Head of department’s signature	prof. Mgr. Petr Páta, Ph.D. Dean’s signature

III. Assignment receipt

The student acknowledges that the master’s thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the master’s thesis, the author must state the names of consultants and include a list of references.

.
Date of assignment receipt

Student’s signature

Acknowledgements

First of all, I want to thank my supervisor, Mgr. Matěj Hoffmann, Ph.D., for the patient guidance through the whole work.

I would like to thank my supervisor-specialist, Ing. Jakub Rozlivek, for continuous suggestions and consultations as well.

Last but not least, I thank prof. Ing. Jiří Matas, Ph.D. for a valuable consultation and ideas in the beginning of the work.

Declaration

I declare that the presented work was developed independently and that I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of university theses.

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských prací.

V Praze dne 18. května 2022

. . . .
Jan Dočekal

Abstract

Human keypoint detection is useful in various applications. Recently, a number of solutions based on deep convolutional neural networks have appeared that recognize human poses in images. Typically, complete human bodies from different distances are detected. For human-robot interaction, human keypoint detection is key to guarantee safe separation distances between the machine and the operator. However, in such close proximity scenarios where only parts of the human body are visible, standard algorithms do not perform well. For human-robot collaboration, robust detection of, for example, only human hands is critical. In this thesis, first, we create and make publicly available a close proximity dataset where only parts of the human body are visible. Second, we quantitatively and qualitatively compare state-of-the-art human keypoint detection methods (OpenPose, MMPose, AlphaPose, Detectron2, and MediaPipe) on this dataset. The results show that the best performing detector is AlphaPose for whole-body annotation and MediaPipe for detection of finger keypoints. Third, we deploy the detector on a humanoid robot iCub with an Intel RealSense RGB-D camera on the head. Detected human keypoints in images are transformed to their 3D positions using depth information from the RealSense camera. We demonstrate the performance in a scenario where the robot uses the detected 3D keypoints for whole-body avoidance maneuvers.

Keywords: human keypoint detection, human hands detection, safe human-robot interaction, 2D to 3D transformation

Supervisor: Mgr. Matěj Hoffmann, Ph.D.

Abstrakt

Detekce částí lidského těla je užitečná pro několik aplikací. V poslední době je k dispozici několik řešení založených na hlubokých konvolučních sítích, které rozpoznají lidské pózy z obrázků. Typicky jsou detekovány celá lidská těla v různých vzdálenostech. Pro interakci člověka s robotem je detekce částí lidského těla klíčem ke garanci bezpečné vzdálenosti mezi strojem a operátorem. Nicméně v případě blízké vzdálenosti, kdy jsou viděny pouze části lidského těla, standardní algoritmy nefungují dostatečně správně. Pro spolupráci člověka s robotem je robustní detekce, například pouze rukou, zásadní. V této práci jsme zaprvé vytvořili a zveřejnili dataset velké blízkosti, kde se objevují pouze části lidského těla. Zadruhé jsme kvantitativně a kvalitativně porovnali nejmodernější metody pro detekci částí lidského těla (OpenPose, MMPose, AlphaPose, Detectron2 a MediaPipe) na tomto datasetu. Výsledky ukazují, že nejlepším detektorem pro anotaci celého těla je AlphaPose a MediaPipe pro detekci prstů. Zatřetí jsme implementovali detekci na humanoidním robotu iCub s Intel RealSense RGB-D kamerou na hlavě. Detekované části lidského těla v obrázcích jsou transformovány do jejich 3D pozic za pomoci hloubkové informace z RealSense kamery. Řešení jsme demonstrovali ve scénáři, kde robot používá detekované 3D části lidského těla pro manévry vyhýbání se celému tělu.

Klíčová slova: detekce částí lidského těla, detekce lidských rukou, bezpečná interakce člověka s robotem, transformace ze 2D do 3D

Překlad názvu: Detekce částí lidského těla z velké blízkosti

Contents

Chapter 1

Introduction

Humanoid robots not only look similarly to humans but we also want them to perceive like humans. Loosely speaking, people have 5 main senses—vision, hearing, touch, taste, and smell. From the point of view of robotics, taste and smell are not beneficial for its behavior. However, touch, hearing, and vision are useful for the desired acting of a robot. One kind of humanoid robot with these senses is iCub [1]. Its body is covered with skin, enabling sensing touch signals. It is also equipped with a microphone to process the sound and cameras located in the eye positions to process images.

This thesis deals with the vision sense of iCub. Computer vision is probably the most complex problem of all human senses. Vision provides a lot of processing, including stereo vision, detection of objects in the observed scene, or orientation in the surroundings. We will focus on the detection task, namely detection of human body keypoints even in close proximity to the robot (camera). A body keypoint is a part of the body that is usually located in the joints of the human skeleton.

The main motivation is the human-robot interaction (HRI), which is desired to be mainly safe. This scenario includes the cooperation of humans and robots, see Figure 1.1. So the crucial task is to know where the human is located in order not to harm the human with robot motion or to obey the orders given by the human. It is desired to be able to do this also with a human in close proximity to the robot and also to detect dense keypoints of human body such as fingers or toes. The method must be primarily robust to fulfill the Speed and separation monitoring [2] in the ISO/TS 15066 norm, which specifies the safety requirements for collaborative robots. The close proximity assumption is taken into account because it is the situation when the touch or harm could be possible. The dense annotation of the human body not only detects the extending part of the human skeleton, but also provides other information, such as gestures made by the human. Human body keypoint detection is also useful for other robots, e.g. robotic arms, from the safety point of view to prevent the harm.

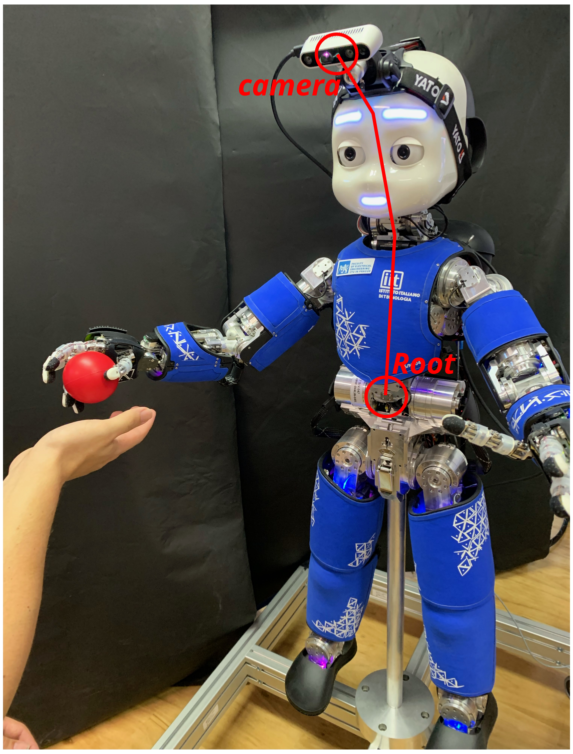


Figure 1.1: iCub humanoid robot with skin during human-robot interaction with highlighted coordinate frames of iCub *Root* frame and Intel RealSense camera frame.

The goals of the thesis are detection of human body parts (keypoints), from which the human body (skeleton) pose can be reconstructed, in the RGB images and transforming those keypoints to 3D position in order to localize the human in the robot working space. The following processing pipeline is visualized in Figure 1.2.

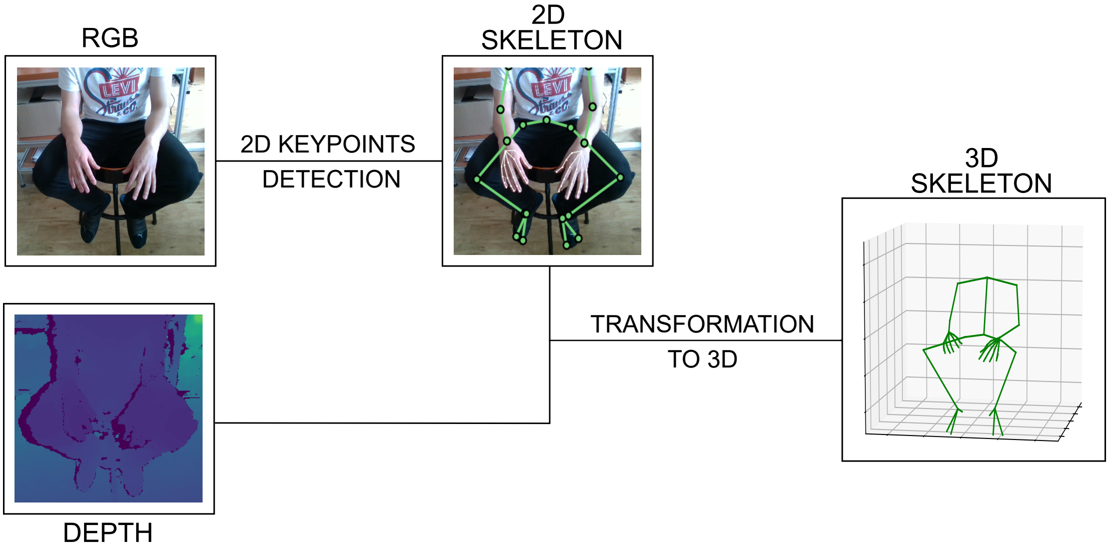


Figure 1.2: Processing pipeline scheme with example inputs and output.

Because the task is very complex and the cameras provided by the original iCub robot provide only low resolution, we will work with iCub equipped with an Intel RealSense camera, see Figure 1.3, which provides both high-resolution images and depth information. The depth information is also much more precise than the one that could be computed from the iCub camera eyes due to mentioned lower resolution and the imperfect position of the eyes obtained from the motor encoders in eyes.

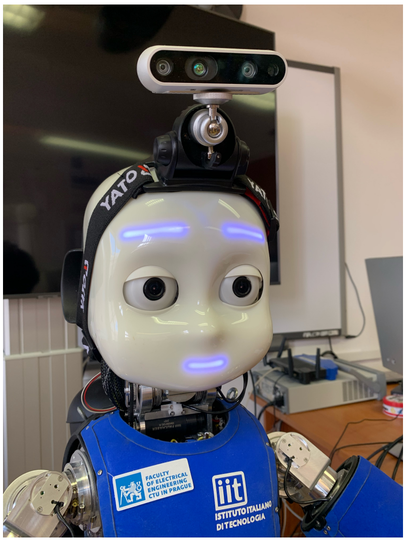


Figure 1.3: iCub humanoid robot with Intel RealSense camera equipped.

Contribution

The main contributions of this thesis are: (i) comparison of detection of dense human keypoints with included finger and face keypoints; (ii) created dataset with people in the close proximity; (iii) pipeline for detection with the focus on the close proximity scenario; (iv) solution implemented for the YARP [3] system.

Structure of the thesis

Firstly, related work will be presented in Chapter 2. Next, the materials and methods, including datasets and evaluation methods description, will be introduced in Chapter 3 alongside with the robot setup. Experiments and their results are described in Chapter 4, with the demonstration of the solution in the HRI scenario in Section 4.5. Finally, the conclusion and discussion of the solution are described in Chapter 5 including a possible future work description.

Chapter 2

Related Work

Recently, there are numerous different approaches to human body detection in RGB or RGB-D images. We can divide them into 2 categories based on how the detected human pose is represented.

Human keypoints detection

There are 2 general approaches for the detection of human body keypoints in 2D images as a 2D human body keypoints. **Bottom-up method** detects the human body keypoints at first and they are reconstructed (connected) to each human body pose (skeleton) afterwards. The bottom-up approach is presented, e.g., in [4–7] or in specific models in the MMPose solutions [8]. The second approach, **top-down method**, proceeds in the reverse order, as they first detect people and their bounding boxes and the desired body keypoints are detected separately in each of the detected human body bounding box. Such approach is presented in [9–11] or in selected models in the MMPose solutions [8] as well. The bottom-up approach is faster than the top-down method with more people appearing in the input images, as the human keypoints are detected all at once.

For the 3D human body keypoints detection in 2D (RGB) images, Cheng et al. [12] conclude with a solution with the focus on self-occlusion, meaning when the human body parts occlude each other. More usual approaches for 3D human detection take into account also the depth from the RGB-D sensors as is presented in [13–16]. Such conclusions correspond the most with the work and the solution presented in this thesis. The approaches in [13–15] combine a 2D human body keypoints detection in the RGB images with fitting a 3D human body model to obtain the resulting 3D keypoints. On the other hand, [16] concludes with a solution detecting the 2D human body keypoints and transforming them directly to the 3D positions based on a disparity map constructed from 2 cameras.

Human 3D surface reconstruction

The human surface representation is a dense human body annotation represented as a human body mesh. A breakthrough solution for this human body annotation is presented in [17–19] and improved in [20]. They conclude with a 3D human body model fitting to a detected human body in a single RGB image referenced as *SMPL* model [17] and *SMPL-X* [20]. Additional works in [21, 22] build on top of the 3D *SMPL* human body model for the human surface detection from a single RGB image. A similar work building on top of the *SMPL-X* model is presented in [23, 24].

In the presented works, the *SMPL* model or *SMPL-X* are directly used in the detection solutions. In DensePose [25] they conclude with a different approach, which does not include the *SMPL* models in the solution. Nevertheless, they also build on top of these models in the data gathering part of the training, as well as the textures provided by the work in [26].

Chapter 3

Materials and Methods

In this chapter we will present the materials and methods used in this thesis. First, we will introduce and describe the existing datasets in Section 3.1 and their contribution to this thesis, followed by the created Close Proximity Dataset in Section 3.2. The human body keypoint detectors will be listed and specified in Section 3.3. We will also describe the iCub robot [1] and the robot setup alongside with the transformation from the Intel RealSense camera frame to the iCub *Root* frame in Section 3.4. The transformation of detected keypoints from 2D to 3D will be presented in Section 3.5. In Section 3.6 we will describe the evaluation processes used in the following experiments in Chapter 4.

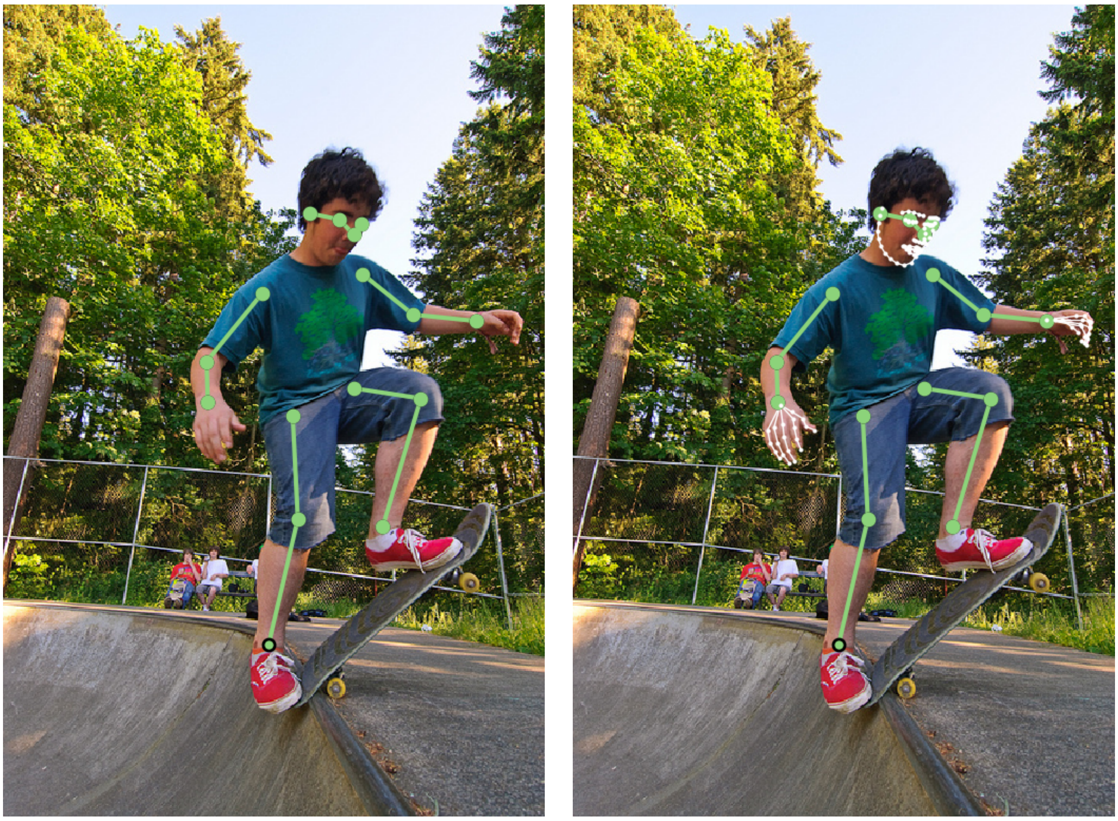
3.1 Existing Datasets

In this section, multiple existing datasets will be introduced together with the description of the advantages or disadvantages following from them. The focus will be on the utilization of the datasets in the close proximity scenario for human body keypoints detection. The keypoint annotation standards are visualized and specified in Section 3.3 in Figure 3.10.

3.1.1 COCO Dataset

One of the most well-known datasets for human keypoints detection is the COCO dataset presented in [27]. It consists of more than 200 000 images with several features, such as object segmentation, object detection, or people with keypoints. The crucial feature for this thesis are the human keypoints annotations.

The original annotations with human keypoints provided with the COCO dataset have 17 keypoints per person. Such annotations do not contain finger, foot, and face keypoints. In [28] the COCO-WholeBody annotations with finger, foot, and face keypoints are presented. All combined together results in 133 keypoints. Especially the finger keypoints are important in the human-robot interaction because it often contains scenarios with handing over some things or other manipulation using fingers. Foot keypoints could be useful because they are an extension of the body which is mandatory for human safety. On the other hand, the face keypoints are not necessary for the safe human-robot interaction because some head keypoints are present in the body annotations as can be seen in Figure 3.1.



(a) Visualization of COCO body and head keypoints.

(b) Visualization of COCO-WholeBody keypoints.

Figure 3.1: Example image from COCO dataset with visualized both original COCO body keypoints and COCO-WholeBody keypoints.

Furthermore, Figure 3.1 shows why finger keypoints are needed, as they significantly extend the length of the arm. It is also noticeable that the COCO-WholeBody annotation is not perfect, as the foot keypoints are not present in the provided example image.

The resulting dataset contains about 64 000 training and 2 700 validation images with people occurring in the scene.

3.1.2 Halpe Dataset

The concurrent work of the COCO dataset with the COCO-WholeBody annotations is the Halpe dataset [9, 29]. The Halpe dataset contains similar human keypoints to the COCO-WholeBody annotations with slight differences. The difference is in the number of keypoints, where the Halpe dataset has 136 keypoints per person. An additional minor difference is in the annotation format.

The Halpe dataset has 41 000 training images with people and shares the same 2 700 validation images with the COCO dataset. An example of the validation image with visualized keypoints is shown in Figure 3.2.

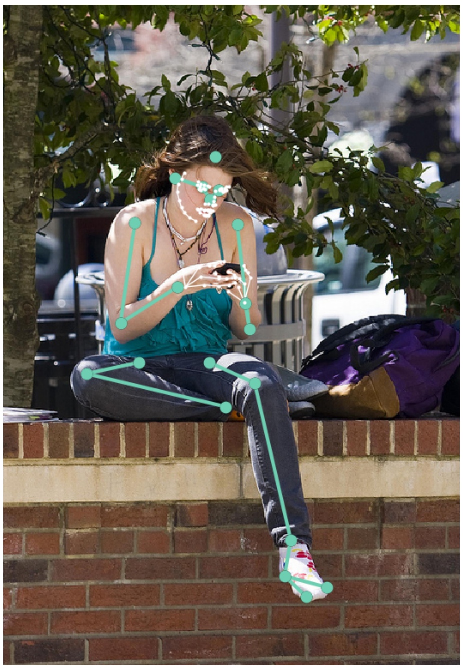


Figure 3.2: Example of validation image from COCO dataset with Halpe keypoints annotation.

3.1.3 DAVIS Dataset

Another dataset has been published in [30] as the DAVIS (Densely Annotated VIdeo Segmentation) dataset. On the contrary to the previous Halpe or COCO datasets, DAVIS dataset consists of Full HD video sequences. This leads to the possibility of improving the human body keypoints detection with tracking.

DAVIS dataset aims at video object segmentation. It means that the video sequences do not contain only people, but also other objects such as animals, vehicles, or sport equipment and activities. Thanks to the last mentioned, sport activities, there are images with people in challenging poses, an example in Figure 3.3.

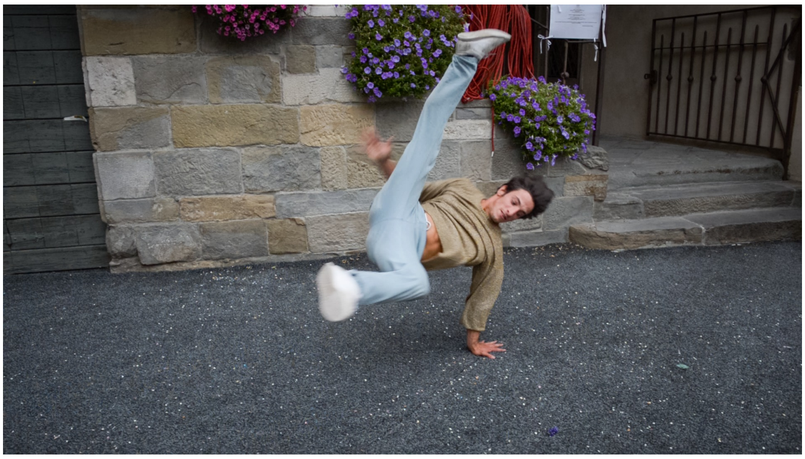


Figure 3.3: Example of video frame from DAVIS dataset "breakdance-flare" video sequence.

Unfortunately, as was stated earlier, the DAVIS dataset annotations contain exclusively object segmentations. That is, they lack the human body pose (keypoints) annotations. From that follows that the DAVIS dataset can not be used for the purposes of this thesis.

3.1.4 DensePose-Posetrack Dataset

The DensePose-Posetrack dataset [31] is another dataset consisting of video sequences with multiple people. The dataset aims at pose estimation and human tracking in videos challenges.

Such challenges are partially similar to the goals of this thesis. The difference is mainly in the close proximity requirements for safe human-robot interaction. Annotations of the video frames contain 17 human body keypoints per person—similarly to the COCO dataset without the WholeBody annotation extension. The absence of finger keypoints is the main disadvantage of the DensePose-Posetrack dataset for possible usage in our approaches and evaluations. An example of a video frame from the dataset is shown in Figure 3.4.

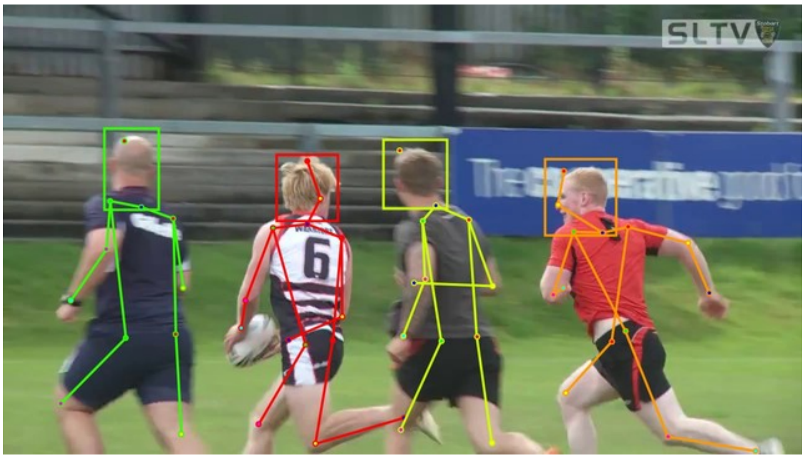


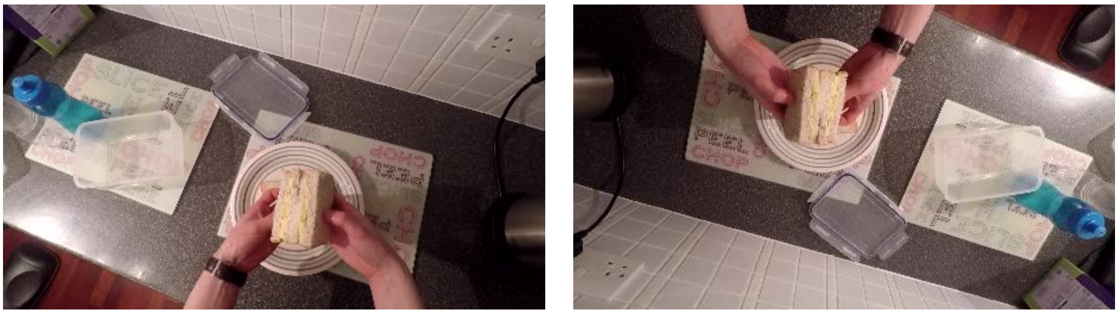
Figure 3.4: Example of image from DensePose-Posetrack dataset with visualized keypoints and head bounding boxes.

3.1.5 EPIC-KITCHENS Dataset

In [32, 33] the EPIC-KITCHENS dataset was published. The dataset contains first-person (egocentric) audio-visual recordings of common human activities in the kitchen. It is the largest egocentric dataset with 100 hours of recordings resulting in about 20 million video frames in Full HD quality.

The first-person view of the human body is not specifically what is desired from the dataset simulating the human-robot interaction scenario. On the other hand, a straightforward transformation of the video frames can be applied—rotating the image by 180 ° leads to a similar view that a robot could observe. This conclusion follows directly from the natural behavior of the human during kitchen duties—looking down on hands or kitchen counter. Figure 3.5 shows an application of this transformation. The rotated image, Figure

3.5b, simulates the situation when the robot observes the action performed by the human interacting with it.



(a) Example of original image from EPIC-KITCHENS dataset.

(b) Rotated example image from EPIC-KITCHENS dataset.

Figure 3.5: Demonstration of the rotation transformation of the image.

The challenges following from the EPIC-KITCHEN dataset are mainly object detection and action detection/recognition. Thus, the respective annotations of videos contain ground truth for action segments along with timestamps of start and end of the action in the corresponding video. Additional automatic annotations provide object/hand masks or bounding boxes.

Even though the EPIC-KITCHENS dataset consists of exclusive and human-robot interaction close views, the need of keypoints annotation is crucial for the purposes of this thesis and that is not provided by this dataset.

Summary of Datasets

Dataset	Keypoints	Data type	Size [frames]
COCO	17	image	200 000
COCO-WholeBody	133	image	66 700
Halpe	136	image	43 700
DAVIS	—	video	3 455
DensePose	17	video	50 000
EPIC-KITCHENS	—	video	20 000 000

Table 3.1: Summary of the properties of presented datasets.

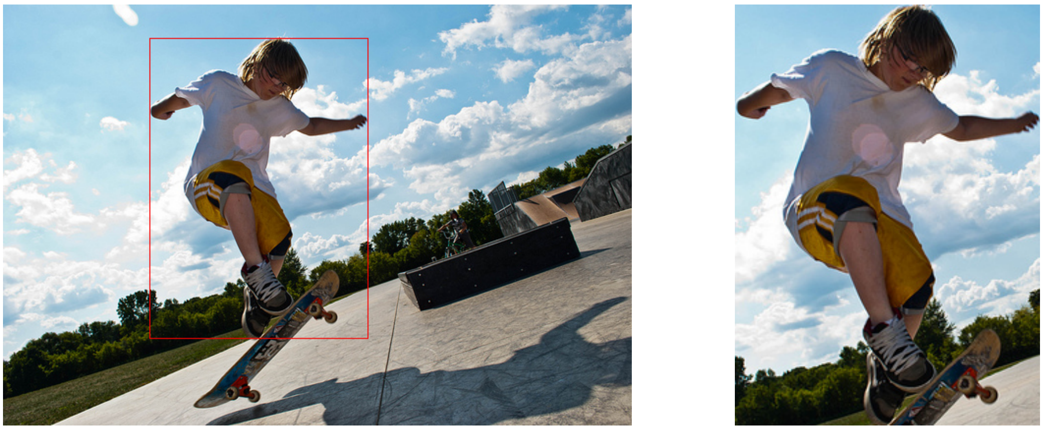
The properties described in this section are summarized in Table 3.1. The DAVIS and the EPIC-KITCHENS datasets contain original video sequences with people occurring, but lack the human body keypoints annotation. The DensePose-Posetrack dataset provides video sequences with the body keypoints but without the finger and the face keypoints as well as the COCO dataset. The COCO dataset is extended with the wholebody keypoints annotation thanks to the COCO-WholeBody dataset. The second dataset with the wholebody keypoints annotation is the Halpe dataset.

3.2 Creating of Close Proximity Dataset

Only the COCO and Halpe datasets, introduced in 3.1.1 and 3.1.2, meet the requirements in terms of dense keypoints annotations. In this section, a method for close proximity scenario simulation will be proposed, and we will contribute with a Close Proximity Dataset with people in close proximity with the wholebody keypoints annotation.

The idea of simulating the close proximity scenario is based on cropping the original images from datasets. Most detectors, which will be introduced in Section 3.3, are trained on the COCO or Halpe dataset. Based on this assumption, some of the detectors could be biased with respect to the training parts of these datasets. Thus, our approach will take into account only the validation part of the COCO and Halpe dataset. As was mentioned earlier, the validation images of both datasets are shared, and therefore none of the detectors is favored due to the training dataset it uses.

The first baseline method of simulating the close proximity focuses on the rough bounding boxes of people occurring in the images. For each annotated person in the image the bounding box is available. Based on this information, each person with its occurrence in the image will be cropped out.



(a) Original image from COCO/Halpe validation dataset with visualized bounding box.

(b) Cropped image according to the bounding box.

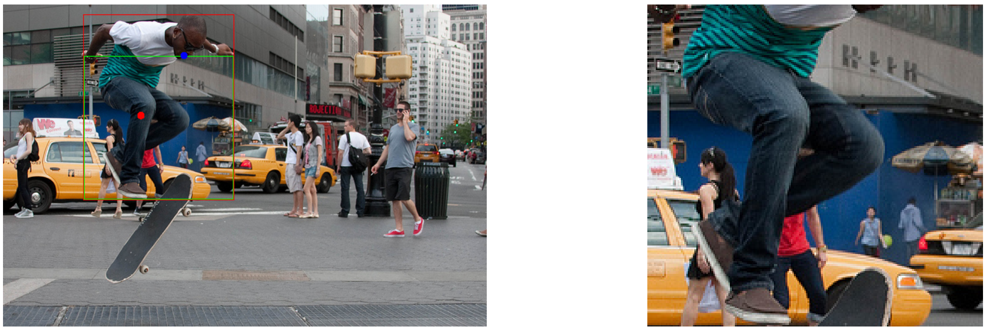
Figure 3.6: Demonstration of cropping the images to simulate closer proximity.

The proposed method is visualized in Figure 3.6. Following this approach, the keypoints annotations are also moved accordingly in order to fit the cropped image correctly.

The main disadvantage of this method is the resulting resolution downscale. The original images often contain multiple people in the background which would lead to very low resolution for each cropped image. An additional condition for the cropped image is to have a number of pixels higher than 20 000.

This baseline method results in the Close Proximity Dataset containing 1624 cropped images with the corresponding dense keypoints annotations.

The second approach to the close proximity simulation extends the first baseline method. The goal is to make the dataset more challenging and to crop out also the head of each person. Finding the head is straightforward, as the head keypoints are usually provided. The 2D position of the rest of the human body with respect to the head is determined as a mean of the rest body keypoints present in the annotation.

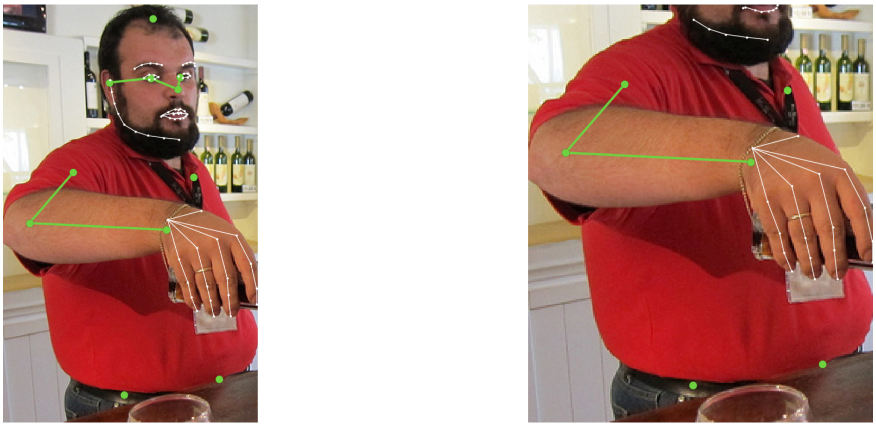


(a) Original image from COCO/Halpe validation dataset with person bounding box (red) and bounding box without head (green) visualized. The head position, body position, is visualized as a blue, red, point respectively.

(b) Cropped image according to the bounding box without head.

Figure 3.7: Demonstration of cropping the images to simulate closer proximity with the head cropped out.

To demonstrate the second approach, Figure 3.7 is provided. The bounding box with head excluded is determined such that the center of body keypoints is included in the bounding box and the head keypoint is on the boarder of the bounding box. The same condition as in the baseline method for the number of pixels in the resulting image is taken into account as well as correction of the keypoint positions in the annotations. Following this extended approach leads to a Close Proximity Dataset with Excluded Head with 1608 images total, an example in Figure 3.8.



(a) Image cropped using the first baseline approach.

(b) Image cropped using the second extended approach.

Figure 3.8: An example image from both Close Proximity Dataset (a) and Close Proximity Dataset with Excluded Head (b) with visualized keypoints.

Both of the Close Proximity Datasets obtained from COCO/Halpe validation part using cropping the images are available at [34] and will be used for further evaluation of 2D human body keypoints detectors in the following part of the thesis.

3.3 Human Keypoint Detectors

There are multiple open source detectors trained for human keypoint detection. They will be introduced and described in this section.

3.3.1 OpenPose

OpenPose [4–7] is the most well-known multi-person system to detect human body keypoints. OpenPose is built using 3 blocks, body and foot detection, hand detection, and face detection. The body and foot detection block is trained on COCO and MPII [35] datasets. Both the hand and face detection blocks use the same approach for training using multiview bootstrapping. OpenPose in its latest version also provides tracking but is limited to a single person.

OpenPose detector is also already part of the `yarpOpenPose` iCub module included in the `human-sensing` repository[1].

3.3.2 Detectron2

More than a single detector, Detectron2 is a library available in [36]. Besides human keypoint detection it also provides solutions to other computer vision tasks such as object detection and segmentation.

Human body keypoint detection included in Detectron2 was trained on the original COCO dataset. This detector outputs only the main body keypoints without hands (fingers). It results in a crucial disadvantage for usage in our approach. Hand detection can be provided externally with some hand speciallized detector, e.g., MediaPipe Hands in Section 3.3.5. Another disadvantage of Detectron2 for human body keypoints detection is the lack of a people tracking feature.

3.3.3 MMPose

Similarly to Detectron2, OpenMMLab project contains several solutions for multiple computer vision problems. MMPose [8] is part of the OpenMMLab project with focus on the human pose estimation.

Unlike Detectron2 keypoint detection, MMPose also contains models for the annotation of the whole human body (including face and finger keypoints). The chosen model is firstly trained on the original COCO dataset and then fine-tuned using the COCO-WholeBody dataset. Furthermore, the MMPose also provides tracking of people poses in the images.

[1] More information available at `https://github.com/robotology/human-sensing`.

3.3.4 AlphaPose

Another whole body human keypoint detector is AlphaPose [9–11]. AlphaPose provides models trained either on the COCO-WholeBody dataset, on the Halpe dataset or the model trained with multi-model knowledge distillation. The last mentioned model is also recommended to use by the authors, as it is ought to be the most accurate and flexible model. As well as MMPose, AlphaPose also contains the possibility of using the tracking of people occurring in the input images.

3.3.5 MediaPipe

MediaPipe is an open source framework from Google introduced in [37]. Alongside with the framework there are also solutions to multiple tasks. The main task corresponding to this thesis is the hand keypoints detection referred to as the MediaPipe Hands and its solution described in [38, 39]. The hand detector provides output in a similar way as in the Halpe or COCO-WholeBody annotations. From this follows that it can be used to obtain alternative finger keypoint detection for other listed detectors or provide the hand detection for detectors which do not detect fingers such as Detectron2 in Section 3.3.2.

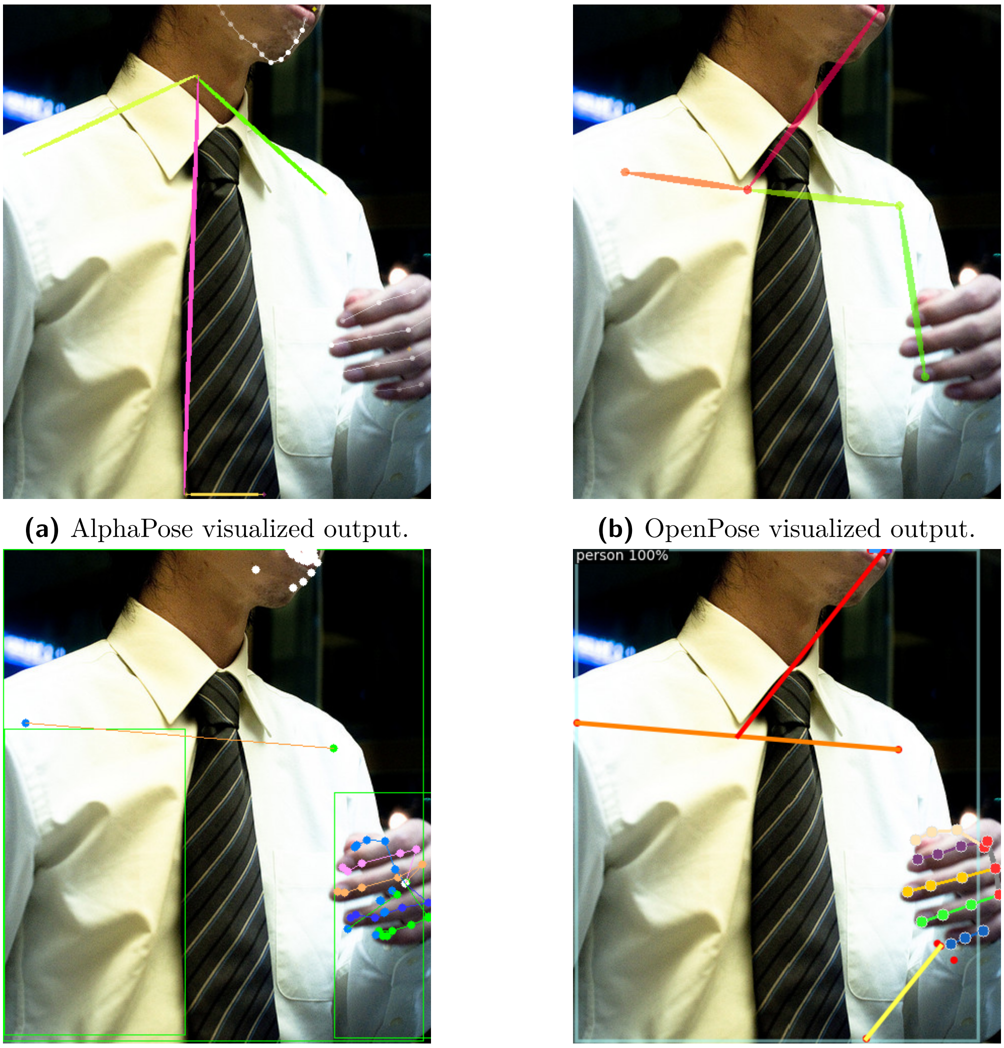
Another solution from MediaPipe can be taken into consideration and it is the Hollistic model. This model combines pose, face, and hand keypoints detection resulting in a dense annotation of human body with 33 body keypoints, 468 face keypoints, and 21 keypoints for each hand. The disadvantage of this solution is that it detects only single person occurring in the image, which is limiting for the real human-robot interaction scenarios.

Summary of Human Keypoint Detectors

Detector	Keypoints	Multi-person tracking	Multi-person detection	Training dataset
OpenPose	135	no	yes	COCO + MPII
Detectron2	17	no	yes	COCO
MMPose	133	yes	yes	COCO + Halpe
AlphaPose	136	yes	yes	Halpe
MediaPipe Hands	21	yes	yes	private
MediaPipe Hollistic	543	no	no	private

Table 3.2: Properties of presented human keypoint detectors.

Each of the presented human keypoint detectors has different properties. These are summarized in Table 3.2. For the purposes of this thesis, MediaPipe Hollistic solution does not suit for the HRI scenario as it is desired to be able to detect multiple persons occurring in the robot working space. The performance of the other detectors will be analyzed in Sections 4.1 and 4.2. As an example, we provide detection by AlphaPose, OpenPose, MMPose, and Detectron2 with MediaPipe for the hand detection in Figure 3.9.



(a) AlphaPose visualized output.

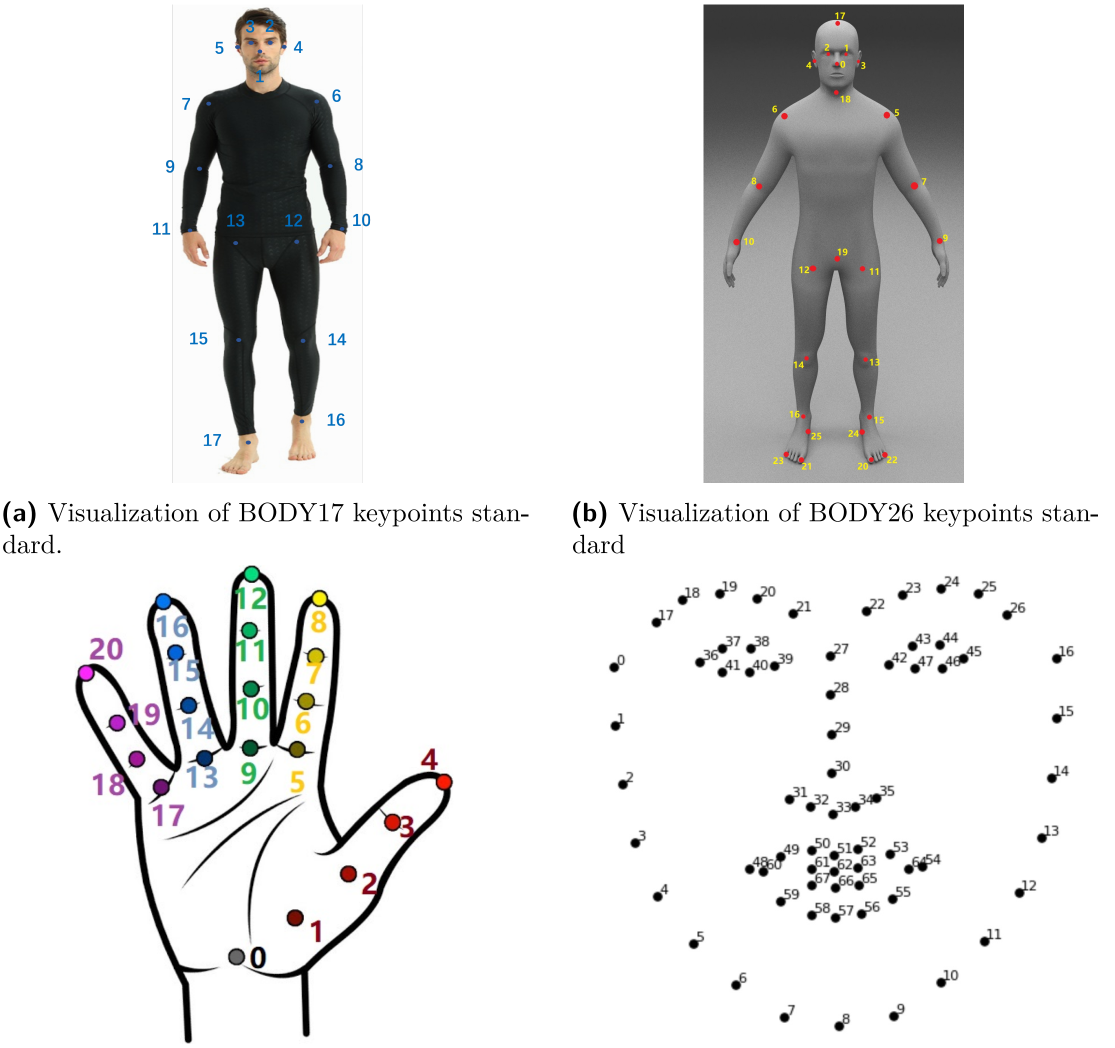
(b) OpenPose visualized output.

(c) MMPose visualized output.

(d) Detectron2 and MediaPipe hand detection visualized output.

Figure 3.9: Human keypoint detectors output on an example image from the Close Proximity Dataset with Excluded Head.

The human body keypoints are defined and visualized in Figure 3.10. The hand keypoints in Figure 3.10c are detected by all detectors in this format for both hands, with the exception of Detectron2, which does not detect hand keypoints at all. As for the face keypoints in Figure 3.10d, this format is standardized for all detectors as well. Only the MediaPipe Hollistic model has even more dense face keypoints with a total of 468. The detectors outputs different formats for the body keypoints. OpenPose detects 25 body keypoints as visualized in Figure 3.10b excluding the keypoint annotated as 17 on the top of the human head. MMPose provides output in the format visualized in Figure 3.10a with foot keypoints from Figure 3.10b (labeled 20-25) in addition. AlphaPose detector uses the BODY26 keypoints standard, Figure 3.10b.



(a) Visualization of BODY17 keypoints standard.

(b) Visualization of BODY26 keypoints standard

(c) Hand keypoints standard.

(d) Face keypoints standard with 68 keypoints.

Figure 3.10: Overview of the wholebody keypoints standards.[2]

3.4 Robot with camera setup

This thesis focuses on the real usage on the iCub humanoid robot [1] with the Intel RealSense D435 camera equipped. RealSense camera provides additional depth information, which is needed for correct 3D coordinates of the detected keypoints. Moreover, the iCub robot is controlled in its base coordinate system named *Root*. Besides transformation the 2D human keypoints to 3D RealSense coordinate system, additional 3D → 3D transformation from the camera coordinate system to the iCub *Root* coordinate system is necessary. A schematic visualization of the iCub coordinate systems with the additional Intel RealSense camera frame is provided in Figure 3.11.

[2] Image 3.10a taken from [28]. Images 3.10b, 3.10c, and 3.10d taken from `https://github.com/Fang-Haoshu/Halpe-FullBody`.

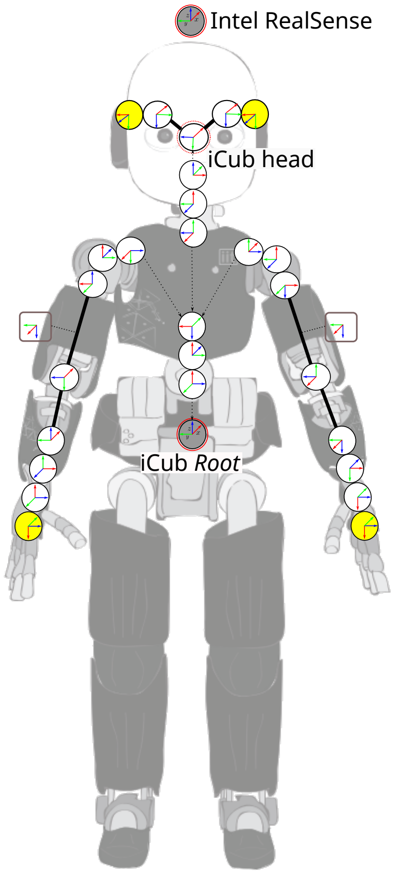


Figure 3.11: iCub humanoid robot coordinate frames with highlighted coordinate frames of iCub *Root* and head frames and Intel RealSense camera frame.

This transformation implies a calibration of the RealSense camera position with respect to the iCub *Root* coordinate system. Because the camera position is not fixed with respect to the *Root* coordinate system, it is solved by calibration of the RealSense camera with respect to some fixed coordinate system, in our case it is the coordinate system of the iCub head. This can be done using the `realsense-holder-calibration` iCub module[3].

After obtaining the RealSense camera position in the iCub head coordinate system, it is possible to compute the transformation from the iCub head coordinate system to the iCub *Root* coordinate system using forward kinematics. The final transformation is then computed as

$$\mathbf{T}_{\text{cam}}^{\text{root}} = \mathbf{T}_{\text{head}}^{\text{root}} \cdot \mathbf{T}_{\text{cam}}^{\text{head}}, \tag{3.1}$$

where $\mathbf{T}_{\text{cam}}^{\text{root}}$ is the resulting transformation from the camera frame to the iCub *Root* frame, $\mathbf{T}_{\text{head}}^{\text{root}}$ is the transformation from the iCub head frame to the iCub *Root* frame obtained with forward kinematics and $\mathbf{T}_{\text{cam}}^{\text{head}}$ is the transformation from the RealSense camera frame to the iCub head frame obtained from the calibration process of the camera. Subsequently, this

[3] Code publicly available at `https://github.com/robotology/realsense-holder-calibration`.

final transformation can be used to transform the 3D position from the camera coordinate system to the iCub *Root* coordinate system.

3.5 Transformation of 2D Keypoints to 3D Keypoints

Human body keypoints are detected in 2D images as 2D positions in the image plane. In order to determine the 3D positions of detected keypoints, we will take into account intrinsic parameters of the camera, such as focal length or principal point position. This transformation is described using the following equation.

$$k \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{3.2}$$

The focal lengths f_x, f_y and the coordinates of the principal point c_x, c_y in Equation 3.2 are the intrinsic parameters of the used camera. 2D position corresponds to the u and v coordinates, and the 3D position is determined as the x, y and z coordinates. The last parameter in Equation 3.2 is the parameter k, which describes the depth of the (u, v) 2D point. From this equation, we can derive formulas for computing the 3D position.

$$x = \frac{k\,(u - c_x)}{f_x} \tag{3.3}$$

$$y = \frac{k\,(v - c_y)}{f_y} \tag{3.4}$$

$$z = k \tag{3.5}$$

The depth k in Equations 3.3, 3.4 and 3.5 is obtained from the used RealSense camera. Following these equations we can compute 3D position of a single pixel (keypoint) in the image.

For better performance of extracting the 3D position, we use a neighborhood of keypoints detected in 2D to compute the final 3D position. The size of the neighborhood is determined dynamically based on the depth of the keypoint and the type of the keypoint. For the body keypoints, e.g. shoulders, elbows or wrists, the neighborhood is chosen with a radius of 2 cm around the keypoint in the 3D coordinate system. Smaller keypoints, such as finger keypoints, have a neighborhood with radius 3 mm. To compute the radius in the 2D pixel coordinates in the image, we derive the expression from Equations 3.3 and 3.4.

$$x_1 = \frac{k\,(u_1 - c_x)}{f_x}$$

$$x_2 = \frac{k\,(u_2 - c_x)}{f_x}$$

$$x_1 - x_2 = \frac{k\,(u_1 - c_x)}{f_x} - \frac{k\,(u_2 - c_x)}{f_x}$$

$$(x_1 - x_2)f_x = k\,(u_1 - c_x) - k\,(u_2 - c_x)$$
$$\frac{f_x}{k}(x_1 - x_2) = u_1 - c_x - u_2 + c_x$$
$$\frac{f_x}{k}(x_1 - x_2) = u_1 - u_2 \tag{3.6}$$

The expression $(x_1 - x_2)$ corresponds to the desired distance in 3D and the expression $(u_1 - u_2)$ is the resulting distance in the 2D image plane on the x-axis. Using the same approach, we can derive Equation 3.7 for the size of the neighborhood on the y-axis in the image plane.

$$\frac{f_y}{k}(y_1 - y_2) = v_1 - v_2 \tag{3.7}$$

The pixels in the derived neighborhood are then transformed to the 3D positions in the camera frame using Equations 3.3, 3.4, 3.5. The final 3D position in the camera frame of the detected 2D keypoint is computed from all the 3D positions of the keypoint in its neighborhood. We propose 2 methods: (i) computing the mean of 3D positions in the neighborhood, and (ii) computing the median of 3D positions from the neighborhood.

Following the described approach, we obtain the 3D position of the detected keypoint in the camera coordinate system. The final step is to compute the 3D position in the *Root* coordinate system of the iCub robot using the transformation from Equation 3.1.

3.6 Evaluation methods

In this section, we will introduce multiple evaluation methods for both 2D keypoint detection and 3D position estimation.

3.6.1 Object Keypoint Similarity

Object Keypoint Similarity (OKS) [40] is a method for evaluating the 2D keypoints detection. It is a similar metric to an Intersection over Union (IoU) used for object detection evaluation. After computing OKS, we can define Average Precision (AP) and Average Recall (AR) in the same manner as for IoU. Object Keypoint Similarity is a commonly used evaluation metric in the COCO keypoint detection challenge.

OKS are computed from the detected keypoints and from the ground truth annotations as follows

$$OKS = \frac{\sum_i \exp\left(\frac{-d_i^2}{2s^2k_i^2}\right)\delta(v_i > 0)}{\sum_i \delta(v_i > 0)}, \tag{3.8}$$

where d_i is the distance between the detected keypoint and its corresponding ground truth position, s is an object (person) scale and k_i is a keypoint constant to control the falloff. δ function is defined as

$$\delta(v_i > 0) = \begin{cases} 1, & \text{if } v_i > 0 \text{ is true} \\ 0 & \text{otherwise,} \end{cases} \tag{3.9}$$

where v_i is the so-called visibility flag of the keypoint which is > 0 if the keypoint occurs in the image, otherwise it is equal to 0. All values correspond to the keypoint with index i and are summed over all ground truth annotated keypoints.

For computing the AP and AR we need to define true/false positive and true/false negative detections. We will use the same metric as in the COCO keypoints challenge. They define thresholds to consider the detection to be true/false positive (or true/false negative). If the computed OKS is greater than a threshold, it is considered as a true positive detection; otherwise it is considered as a false positive.

When we have obtained the number of true/false positives and negatives, we can compute average precision and average recall from Equations 3.10 and 3.11, where TP is the number of true positives, FP is the number of false positives and FN is the number of false negatives.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{3.10}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3.11}$$

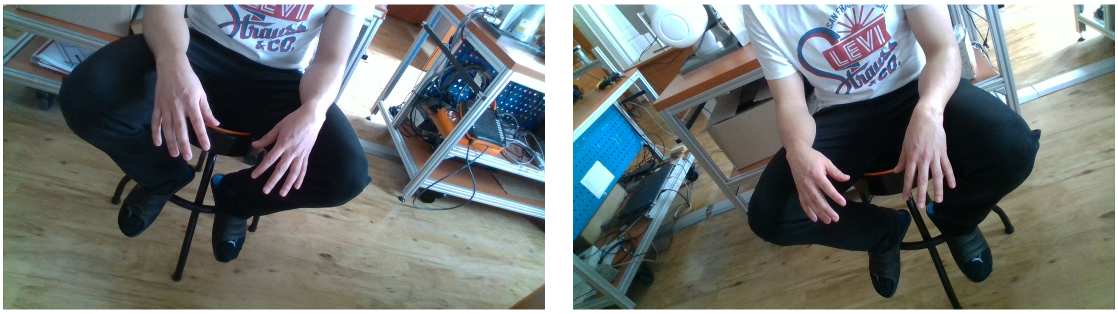
We used the same OKS thresholds as in the COCO challenge which are: 0.5, 0.75 and averaging the precision and recall over thresholds from the interval from 0.5 to 0.95 with a step value 0.05. This means that we sample the interval with the mentioned step value and compute precision and recall for all these threshold values and the resulting average precision (AP) and average recall (AR) over this interval of thresholds are the mean of computed precisions and recalls.

3.6.2 3D Positions Evaluation

In the ideal scenario, we would like to evaluate our estimated 3D positions by comparing with the actual positions. Determining the ground truth 3D positions of the human keypoints with respect to the robot *Root* frame is a problem due to an unknown exact position of the *Root* coordinate frame, meaning that we do not know how to measure the human position with respect to the *Root* frame, and problematic measuring of the position with respect to the roughly estimated *Root* frame. We propose 2 experiments to solve these issues.

Relative Position Evaluation

The first experiment for 3D positions evaluation assumes a static human pose with respect to the *Root* frame. During this procedure, the robot will observe the human pose from different positions but with the static *Root* frame. Specifically, the robot will move with its torso and head parts in order to observe the human keypoints positions from 100 predefined positions. The example of images recorded with the Intel RealSense camera is provided in Figure 3.12. The desired output is ought to output the same 3D positions of detected human body keypoints for all of the robot poses.



(a) Example input image from one of the predefined robot positions.

(b) Example input image from another of the predefined robot positions.

Figure 3.12: Example input images recorded with Intel RealSense camera during experiment for the relative position evaluation.

The detected 2D positions of the keypoints are transformed to the Intel RealSense camera frame using the method described in Section 3.5 using both mean and median approach. These 3D positions are transformed to the robot *Root* frame as described in Section 3.4. The resulting 3D positions of the human body keypoints correspond to the static iCub *Root* coordinate frame.

This approach is able to overcome the lack of ground truth 3D positions and results can be evaluated relatively thanks to static position of observed human. It is highly inspired by camera calibration processes where the approach is similar.

Absolute Position Evaluation

The second experiment aims at the comparison of the 3D positions with respect to the reference frame, located in a different position than the robot coordinate system, and the estimated 3D positions provided by our pipeline with 2D keypoints detection and transformation to the 3D positions in the camera frame. As was mentioned earlier, we do not have the ground truth 3D position of human keypoints. In fact, we can obtain their estimation using the Qualisys Motion Capture (MoCap) System, where the crucial keypoints can be marked with special marker points which are then located using the system. The specifications of the Qualisys MoCap System are 8× Miqus M3 cameras with 2 MP (1824×1088), 340 Hz in full resolution (0.5 MP, 650 Hz in the reduced resolution) with 1× Miqus Video camera, 2 MP (1920×1080), 85 Hz in full resolution (1 MP, 180 Hz in the reduced resolution).

Thanks to the MoCap system, the observed human can move during the experiment. This results in the ability to compare the detected 3D positions of human body keypoints with its estimated ground truth positions. Such an approach yields another problem, which is obtaining the transformation from the Intel RealSense to the reference frame of the Qualisys MoCap system. The absolute position evaluation experiment is visualized in Figure 3.13 alongside with the transformation we seek.

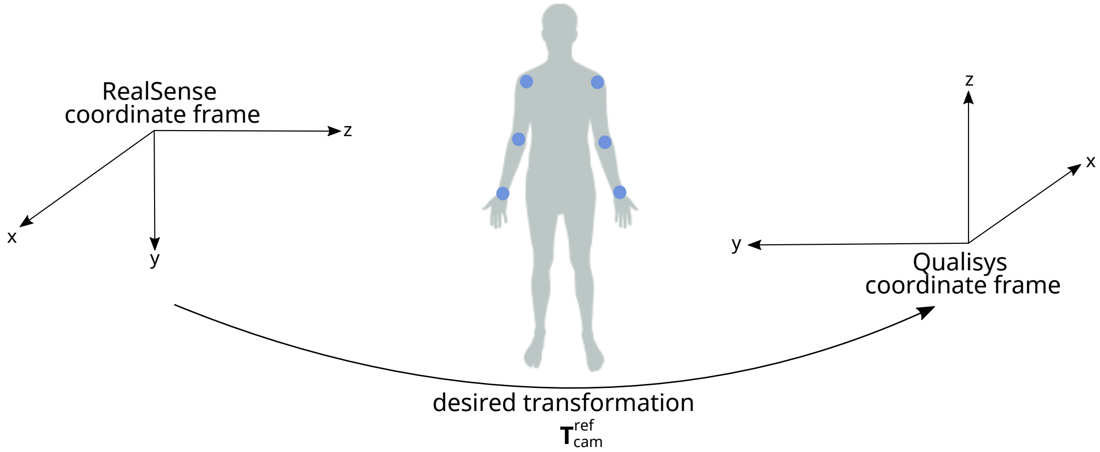


Figure 3.13: Demonstration of the absolute position evaluation experiment with the Intel RealSense coordinate frame, Qualisys reference coordinate frame and desired transformation $\mathbf{T}_{\text{cam}}^{\text{ref}}$.

For determining the transformation from the Intel RealSense camera to the reference frame of Qualisys we used an algebraic approach using the Singular Value Decomposition (SVD) for estimating the rotation part of the transformation [41]. Let us assume we have N points in both coordinate frames (camera frame and reference frame of Qualisys) with their correspondences. The points in the Intel RealSense camera frame were manually annotated in the 2D image to precisely determine the positions of the marked points and transformed to the 3D camera frame using the method described in Section 3.5. We organize those points into two matrices

$$\mathbf{A} = \begin{bmatrix} x_{1_A} & y_{1_A} & z_{1_A} \\ x_{2_A} & y_{2_A} & z_{2_A} \\ \vdots & \vdots & \vdots \\ x_{\mathrm{N}_A} & y_{\mathrm{N}_A} & z_{\mathrm{N}_A} \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} x_{1_B} & y_{1_B} & z_{1_B} \\ x_{2_B} & y_{2_B} & z_{2_B} \\ \vdots & \vdots & \vdots \\ x_{\mathrm{N}_B} & y_{\mathrm{N}_B} & z_{\mathrm{N}_B} \end{bmatrix}$$

where x_{i_A}, y_{i_A}, z_{i_A} are the xyz coordinates in the first (Intel RealSense camera) coordinate system of the i-th point. The corresponding point in the second (Qualisys reference) coordinate system is as well on the i-th row of the $\mathbf{B}$ matrix.

The next step is to compute centroids of both sets of points which is a mean of all columns in the corresponding matrix. When we subtract the centroid from all rows of the matrix we end up with a set of points with centroid located in the origin (zero) position. We will denote this matrix $\mathbf{A}_{\mathrm{c}}$, $\mathbf{B}_{\mathrm{c}}$ respectively.

Now we can compute the covariance matrix $\mathbf{H}$ of both sets of points, Equation 3.12, and find the SVD factorization, Equation 3.13.

$$\mathbf{H} = \mathbf{A}_{\mathrm{c}}^{T}\mathbf{B}_{\mathrm{c}} \tag{3.12}$$

$$\mathbf{USV}^{T} = \mathbf{H} \tag{3.13}$$

The final rotation part of the wanted transformation from the camera frame to the MoCap reference frame is computed from the orthogonal matrices of the SVD factorization in Equation 3.14.

$$\mathbf{R} = \mathbf{VU}^{T} \tag{3.14}$$

Finding the translation part of the transformation is straightforward. This method is described in Equation 3.15, where $\mathbf{c}_A$, $\mathbf{c}_B$, are the centroids of the set $\mathbf{A}$ and $\mathbf{B}$, respectively.

$$\mathbf{t} = -\mathbf{R}\mathbf{c}_A + \mathbf{c}_B \tag{3.15}$$

The transformation $\mathbf{T}_{\mathrm{cam}}^{\mathrm{ref}}$ from the camera frame to the reference frame of the Qualisys Motion Caption System is standardly constructed as follows

$$\mathbf{T}_{\mathrm{cam}}^{\mathrm{ref}} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix}. \tag{3.16}$$

Thanks to this transformation $\mathbf{T}_{\mathrm{cam}}^{\mathrm{ref}}$ we are able to determine the 3D positions of the detected keypoints and compare them with the 3D positions tracked with the Qualisys Motion Capture system. For further evaluation we will use the Mean Absolute Error (MAE), Equation 3.17, in each of the axis during the time of motion to determine the precision of 2D human body keypoints detections combined with transformation to the 3D positions.

The formula for Mean Absolute Error is as follows

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n} |x_i - \hat{x}_i|}{n}, \tag{3.17}$$

where x_i is the coordinate of the detected and transformed 3D keypoint and $\hat{x}_i$ is the ground truth coordinate estimated by the Qualisys Motion Capture system.

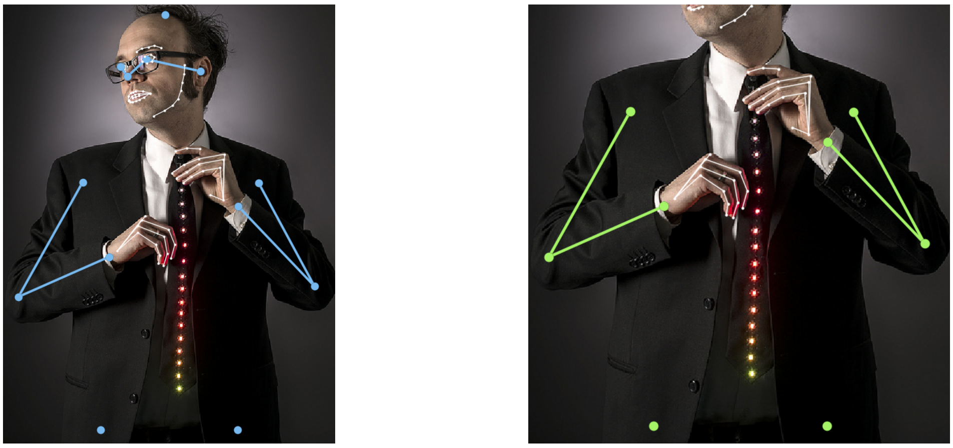
Chapter 4

Experiments and Results

This chapter contains the description of the performed experiments and the results. All the human body keypoints detectors introduced in Section 3.3 are analyzed and evaluated on the Close Proximity Datasets in Section 4.1. The best performing detectors are also analyzed in the real scenarios of human-robot interaction (HRI) in Section 4.2. The experiments and results of the 3D positions of human body keypoints are presented in Sections 4.3 and 4.4. In the last Section 4.5 we will introduce a demonstration of the proposed method in a human-robot interaction scenario.

4.1 Evaluation of Detectors on the Close Proximity Datasets

Both the Close Proximity Dataset and the Close Proximity Dataset with Excluded Head, Section 3.2, will be annotated using all detectors described in Section 3.3. An example of input images from the evaluated datasets is provided in Figure 4.1. These annotations will be evaluated using Average Precision (AP) and Average Recall (AR) metric using Object Keypoint Similarity (OKS) presented in Section 3.6.1. The AP and AR will be listed for 4 different parts of the human body—*Body*, *Foot*, *Hand* and *Wholebody*.



(a) Example image from Close Proximity Dataset.

(b) Example image from Close Proximity Dataset with Excluded Head.

Figure 4.1: Example images from both Close Proximity Dataset (a) and Close Proximity Dataset with Excluded Head (b) with visualized ground truth annotation of human keypoints.

Body parts	Average Precision			Average Recall		
	@0.5:0.95	@0.5	@0.75	**@0.5:0.95**	@0.5	@0.75
Body	0.683	0.875	0.778	0.756	0.927	0.830
Foot	0.506	0.612	0.523	0.762	0.858	0.783
Hand	0.108	0.301	0.054	0.224	0.498	0.169
Wholebody	0.296	0.779	0.419	0.508	0.839	0.531

Table 4.1: AlphaPose evaluated on the Close Proximity Dataset.

Body parts	Average Precision			Average Recall		
	@0.5:0.95	@0.5	@0.75	**@0.5:0.95**	@0.5	@0.75
Body	0.593	0.779	0.628	0.707	0.874	0.739
Foot	0.550	0.629	0.560	0.791	0.863	0.802
Hand	0.118	0.287	0.082	0.244	0.464	0.223
Wholebody	0.264	0.589	0.206	0.423	0.730	0.397

Table 4.2: AlphaPose evaluated on the Close Proximity Dataset with Excluded Head.

Tables 4.1 and 4.2 show the resulting AP and AR on both Close Proximity Datasets for AlphaPose detector. *Body* part includes the main skeleton joints such as shoulders, elbows or wrists, and the head keypoints such as nose or ears. *Foot* part has only 6 keypoints in total, 3 per foot. Finger keypoints are included in the *Hand* body part and all these keypoints are taken into account in the *Wholebody* part. Face keypoints which are also detected by AlphaPose were not mentioned as they are not significant for the purposes of this thesis.

Average Precision (AP) @0.5:0.95 means an average precision over OKS with thresholds set from the interval from 0.5 to 0.95 with step value 0.05 and, for example, AR @0.5 stands for average recall using OKS with threshold set to 0.5, as previously described in Section 3.6.1. The most important body parts for us are the *Body* and *Hand* parts, as they will be the most common in human-robot interaction scenarios.

Table 4.2 shows that the Close Proximity Dataset with Excluded Head is harder to annotate because the AP (AR) is lower for the body part. On the other hand, the detection of finger keypoints is similar for both datasets.

Body parts	Average Precision			Average Recall		
	@0.5:0.95	@0.5	@0.75	**@0.5:0.95**	@0.5	@0.75
Body	0.291	0.508	0.289	0.579	0.849	0.631
Foot	0.129	0.172	0.129	0.591	0.696	0.594
Hand	0.077	0.167	0.064	0.314	0.524	0.328
Wholebody	0.142	0.316	0.126	0.391	0.667	0.422

Table 4.3: OpenPose evaluated on the Close Proximity Dataset.

Body parts	**Average Precision**			**Average Recall**		
	@0.5:0.95	@0.5	@0.75	**@0.5:0.95**	@0.5	@0.75
Body	0.235	0.391	0.222	0.496	0.668	0.497
Foot	0.143	0.189	0.143	0.560	0.650	0.565
Hand	0.089	0.168	0.082	0.321	0.476	0.331
Wholebody	0.134	0.263	0.124	0.360	0.548	0.380

Table 4.4: OpenPose evaluated on the Close Proximity Dataset with Excluded Head.

OpenPose results are summarized in Tables 4.3 and 4.4. In comparison with the AlphaPose detector (4.1, 4.2), the OpenPose detection algorithm is outperformed in all the body parts.

Body parts	**Average Precision**			**Average Recall**		
	@0.5:0.95	@0.5	@0.75	**@0.5:0.95**	@0.5	@0.75
Body	0.711	0.846	0.782	0.851	0.967	0.906
Foot	0.440	0.525	0.458	0.793	0.879	0.821
Hand	0.138	0.427	0.042	0.302	0.695	0.226
Wholebody	0.352	0.718	0.274	0.533	0.869	0.526

Table 4.5: MMPose evaluated on the Close Proximity Dataset.

Body parts	**Average Precision**			**Average Recall**		
	@0.5:0.95	@0.5	@0.75	**@0.5:0.95**	@0.5	@0.75
Body	0.653	0.804	0.690	0.815	0.940	0.844
Foot	0.421	0.490	0.434	0.801	0.865	0.817
Hand	0.249	0.526	0.208	0.430	0.756	0.435
Wholebody	0.387	0.684	0.380	0.563	0.847	0.589

Table 4.6: MMPose evaluated on the Close Proximity Dataset with Excluded Head.

MMPose wholebody keypoint detector provides results competitive with the AlphaPose detector. The AP and AR are summarized in Tables 4.5 and 4.6. The average precisions of *Body* part and the *Hand* part are higher in comparison with AlphaPose detection and these parts are the most important for us. Especially the difference in the hand keypoints detection is a valuable advantage.

Body parts	**Average Precision**			**Average Recall**		
	@0.5:0.95	@0.5	@0.75	**@0.5:0.95**	@0.5	@0.75
Body	0.671	0.858	0.734	0.765	0.930	0.816
Hand	0.118	0.307	0.082	0.243	0.505	0.217

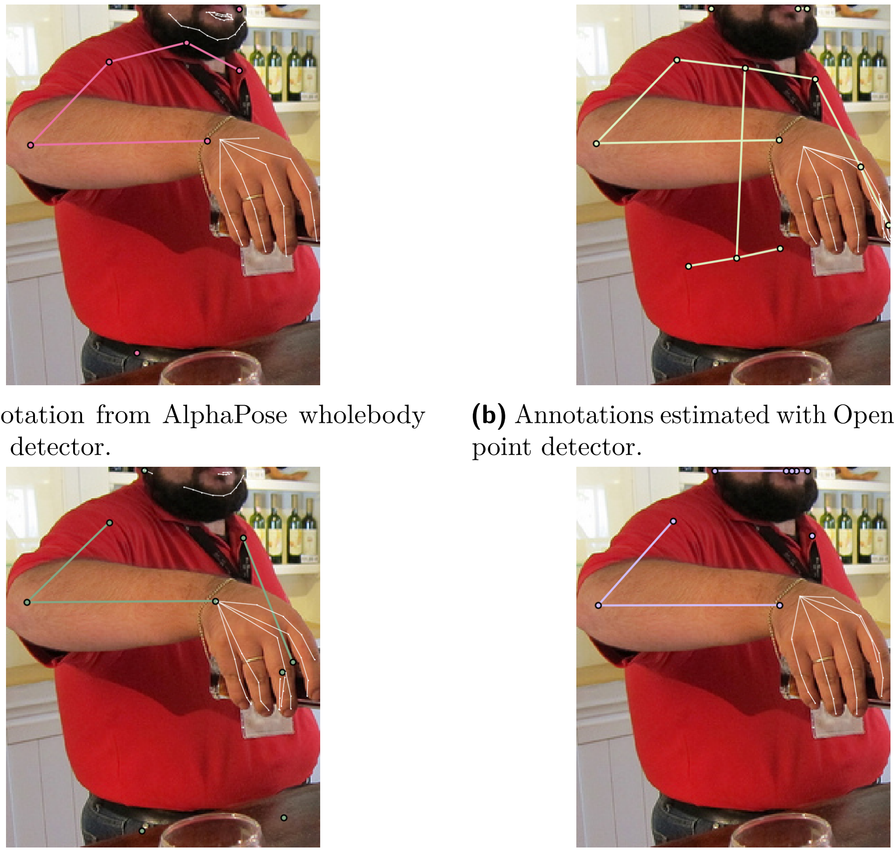
Table 4.7: Detectron2 in combination with MediaPipe hand detector evaluated on the Close Proximity Dataset.

Body parts	**Average Precision**			**Average Recall**		
	@0.5:0.95	@0.5	@0.75	**@0.5:0.95**	@0.5	@0.75
Body	0.572	0.783	0.601	0.696	0.888	0.726
Hand	0.113	0.259	0.093	0.268	0.481	0.257

Table 4.8: Detectron2 in combination with MediaPipe hand detector evaluated on the Close Proximity Dataset with Excluded Head.

The last evaluation on the Close Proximity Datasets is done on 2 detectors simultaneously. For the *Body* part detection the Detectron2 human body keypoint detector is used. Detectron2 is combined with MediaPipe hand detection, which provides annotations for the *Hand* part. The *Foot* keypoints are not present in neither Detectron2 or MediaPipe detection and therefore are not evaluated. In addition, the evaluation of *Wholebody* would be biased due to the lack of *Foot* key points.

Resulting AP and AR are listed in Tables 4.7 and 4.8. Both body parts provide similar results when compared with AlphaPose or MMPose wholebody keypoint detection.



(a) Annotation from AlphaPose wholebody keypoint detector.

(b) Annotations estimated with OpenPose keypoint detector.

(c) MMPose wholebody annotations.

(d) Annotations of body keypoints by Detectron2 and hand keypoints detected with MediaPipe hand detector.

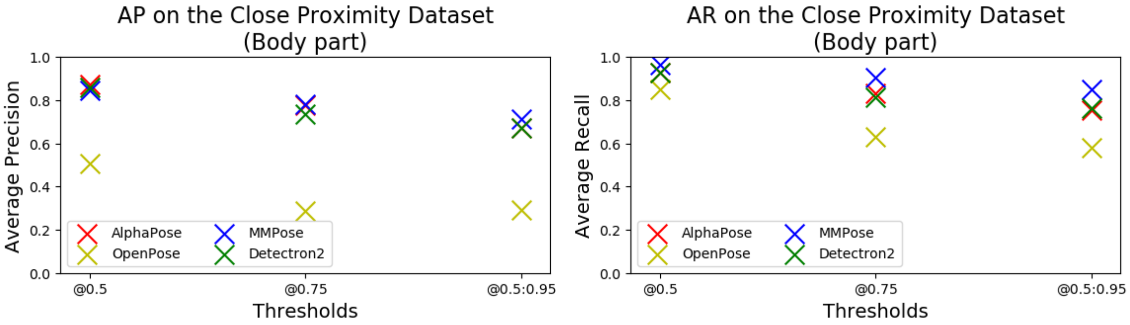
Figure 4.2: Visualized human body keypoint annotations on an example image from Close Proximity Dataset with Excluded Head.

We also provide example annotations from the detectors on an example image from the Close Proximity Dataset with Excluded Head in Figure 4.2. The original image with the ground truth annotation is shown in Figure 3.8b.

From Figure 4.2, it is visible that OpenPose is outperformed by other detectors not only statistically, as the hip keypoints are detected on the abdomen in Figure 4.2b. The hip keypoints are detected by AlphaPose in Figure 4.2a and by MMPose 4.2c.

Summary

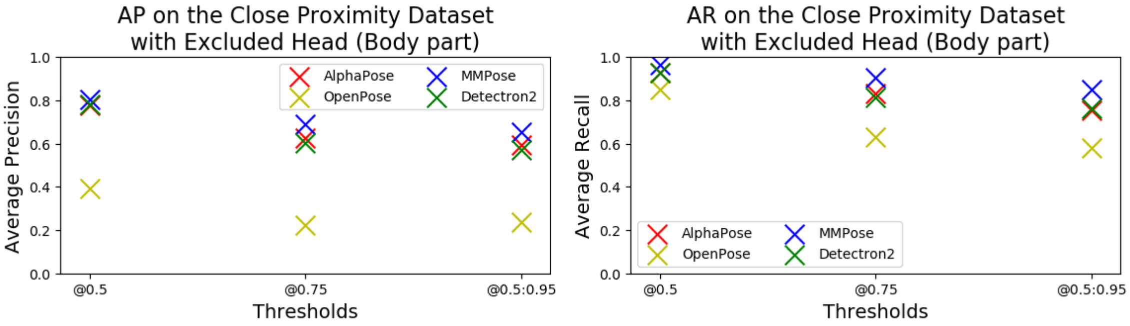
The AlphaPose and MMPose are considered to be the best performing detectors. MediaPipe hand detector also provides correct annotations for finger keypoints. Detectron2 performs well too, but there is the disadvantage of missing finger, foot, and face annotations, and thus we can choose the AlphaPose or MMPose with similar results but also with finger, face, and foot keypoints included.



(a) Average Precision of the detectors on the *Body* part.

(b) Average Recall of the detectors on the *Body* part.

Figure 4.3: Graphical visualization of the results of all detectors for the *Body* part on the Close Proximity Dataset.



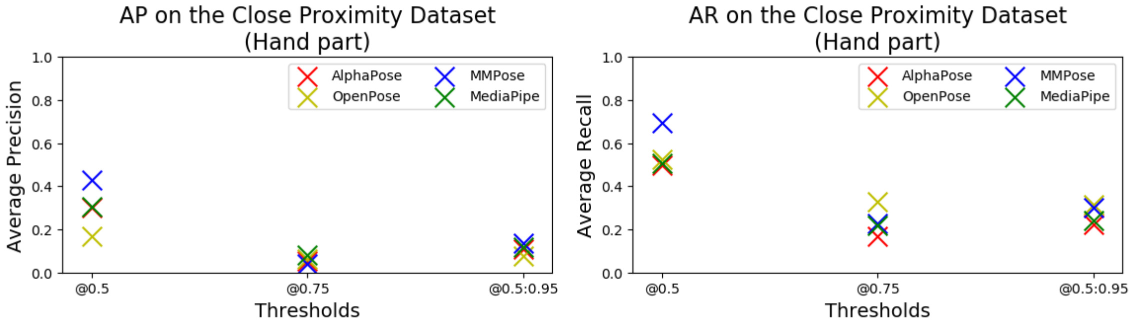
(a) Average Precision of the detectors on the *Body* part.

(b) Average Recall of the detectors on the *Body* part.

Figure 4.4: Graphical visualization of the results of all detectors for the *Body* part on the Close Proximity Dataset with Excluded Head.

Additionally, we provide a graphical comparison of the detectors with the Average Precisions and Recalls from the tables in this section in Figures 4.3, 4.4, 4.5 and 4.6.

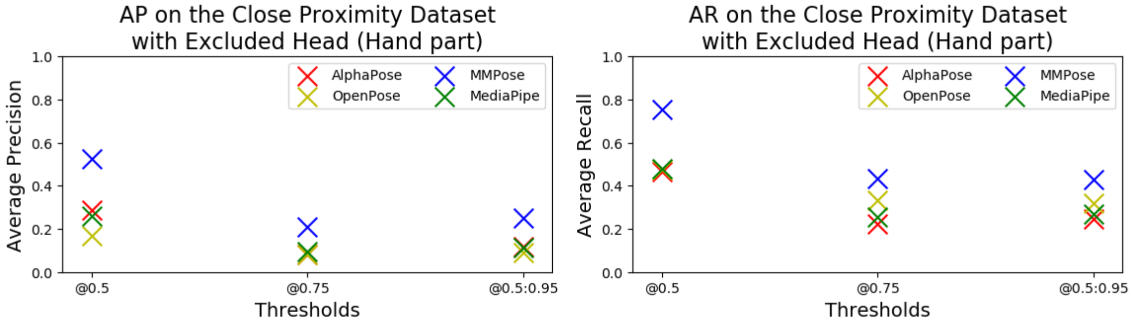
As we mentioned earlier, all the detectors outperform the OpenPose human keypoints detection significantly in the *Body* keypoints part on both of the created Close Proximity Datasets. The difference is visible in the example annotation Figure 4.2 and in Figures 4.3 and 4.4 in the comparison with other detectors.



(a) Average Precision of the detectors on the *Hand* part.

(b) Average Recall of the detectors on the *Hand* part.

Figure 4.5: Graphical visualization of the results of all detectors for the *Hand* part on the Close Proximity Dataset.



(a) Average Precision of the detectors on the *Hand* part.

(b) Average Recall of the detectors on the *Hand* part.

Figure 4.6: Graphical visualization of the results of all detectors for the *Hand* part on the Close Proximity Dataset with Excluded Head.

Finally, we provide a Precision-Recall graphs for both *Body* and *Hand* parts in Figure 4.7—the AP and AR correspond to the threshold interval from 0.5 to 0.95 with a step value 0.05. The hand keypoints detection, Figures 4.5, 4.6, 4.7b, is a crucial problem for all of the detectors due to the task complexity and also due to the resolution reduction in the datasets images. Neither of the detectors performs well enough to pick the best one for the hand detection. We will focus on this problem in the following section with analysis on the real data.

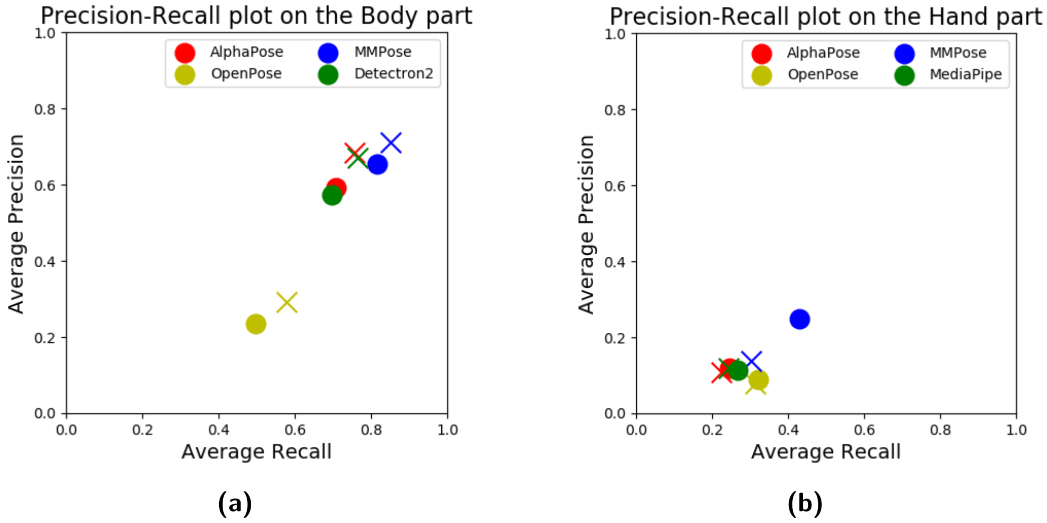
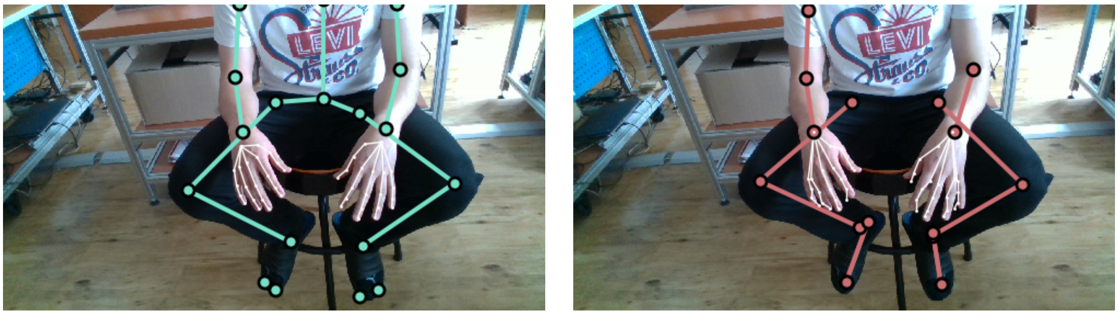


Figure 4.7: Precision-Recall graphs for both *Body* and *Hand* parts with AP and AR @0.5:0.95. Colors correspond to the detector; × marker stands for the Close Proximity Dataset; ● marker stands for the Close Proximity Dataset with Excluded Head.

4.2 Analysis of Detectors on Real Data

Based on the results from the previous Section 4.1, we implemented the 2D detection of human body keypoints in the YARP (Yet Another Robot Platform) [3] system running on the real iCub robot. We took into account AlphaPose and MMPose wholebody detection in combination with MediaPipe hand detector, as they provided the best results on the Close Proximity Datasets in the wholebody keypoints annotation, and hand keypoints annotation, respectively.

Because of complicated and time inefficient manual annotating of real images recorded on the Intel RealSense camera on the iCub robot, we analyzed the data by observation of detected human body keypoints qualitatively only. We focus on the correct detection of keypoints alongside with robustness of the annotation and also speed of the detection.



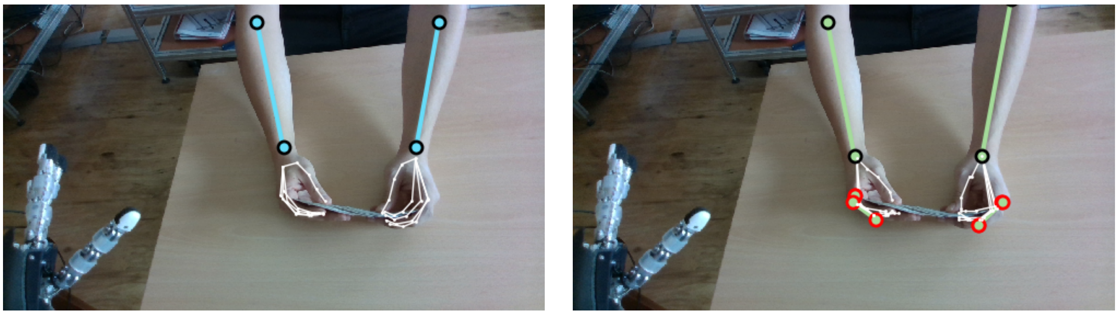
(a) Human body keypoints detected with AlphaPose detector.

(b) Human body keypoints detected using MMPose detector.

Figure 4.8: An example real data image with detection of human keypoints using both AlphaPose and MMPose.

Figure 4.8 shows the different detections by AlphaPose (4.8a) and MMPose (4.8b). We can notice the difference mainly in the finger keypoints, where AlphaPose detects them smoothly and more precisely than MMPose. MMPose detection on fingers provides keypoints in an unnatural manner, as the fingers look like they were broken and often even outside the correct position of the finger. It would cause problems in 3D position computation due to incorrect depth information for such keypoints.

Another issue with MMPose detection is that it provides more false positive detections when only a part of the human body is visible. This is demonstrated in Figure 4.9. AlphaPose detection, Figure 4.9a, detects precisely the visible human keypoints but MMPose additionaly outputs false positive detection of foot keypoints, which are detected on the hands and annotated with red circle points (keypoints) in Figure 4.9b.

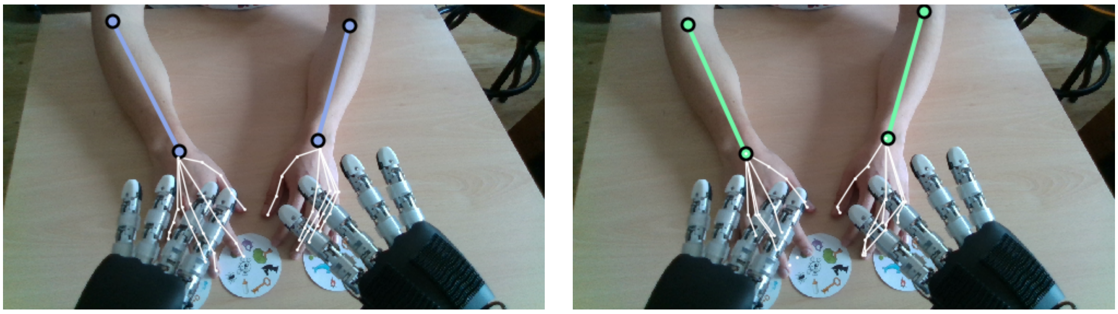


(a) Human body keypoints detected with AlphaPose detector.

(b) Human body keypoints detected using MMPose detector.

Figure 4.9: Demonstration of false positive detection (red circled keypoints) with MMPose human body keypoints detector.

We also simulated a scenario where human hands are partially occluded by iCub robot parts. It is important that the detectors do not detect iCub as a human, as it would lead to incorrect behavior in future usage, e.g. avoiding the human during motion. The image example alongside the detections is visualized in Figure 4.10.



(a) Human body keypoints detected with AlphaPose detector.

(b) Human body keypoints detected using MMPose detector.

Figure 4.10: Scenario with covering the human body parts with iCubs hands.

As can be seen in Figure 4.10, both detections by AlphaPose and MMPose do not consider iCub hands as a human body, which is mandatory for later usage. The detection itself is pretty similar by both as the finger keypoints which are barely visible are not detected precisely by neither of the detectors.

Detector	Speed [fps]
AlphaPose	15
MediaPipe	15
MMPose	7-8

Table 4.9: Speed of AlphaPose, MediaPipe and MMPose keypoint detectors in fps. Measured on PC with Intel Xeon W-2295 CPU (36 × 3.0 GHz) and NVIDIA Quadro RTX 5000 GPU (16GB GDDR6).

Another difference between the chosen detectors is in the time of detection. AlphaPose and MediaPipe hand detection are able to run at approximately 15 fps. On the contrary, MMPose detection is almost 2 times slower and runs only at about 7-8 fps. The time requirements are measured on the PC with Intel Xeon W-2295 CPU (36 × 3.0 GHz) and NVIDIA Quadro RTX 5000 GPU (16GB GDDR6) and are listed in Table 4.9. The speed factor is also important for real usage because we want to know the position of a human as often as possible in order to prevent any harm which could be caused to a human.

Based on the mentioned advantages and disadvantages of chosen detectors we propose a method that takes into account both AlphaPose and MediaPipe detectors as their speed of detection are almost similar and MMPose detection does not provide any significant improvement and is slower. Our approach will detect the human body in the input image using AlphaPose detector in combination with the MediaPipe hand detection, see Figure 4.11 with the pipeline visualization. MediaPipe is able to detect finger keypoints precisely even in challenging poses, but does not detect fingers when only part of the hand (fingers) is visible. On the other hand, AlphaPose is able to detect such partially observable fingers thanks to the additional information of the wrist and other hand keypoints. The conclusion is that when MediaPipe detects finger keypoints, we will consider them preferably and replace them in the whole body detection provided by AlphaPose.

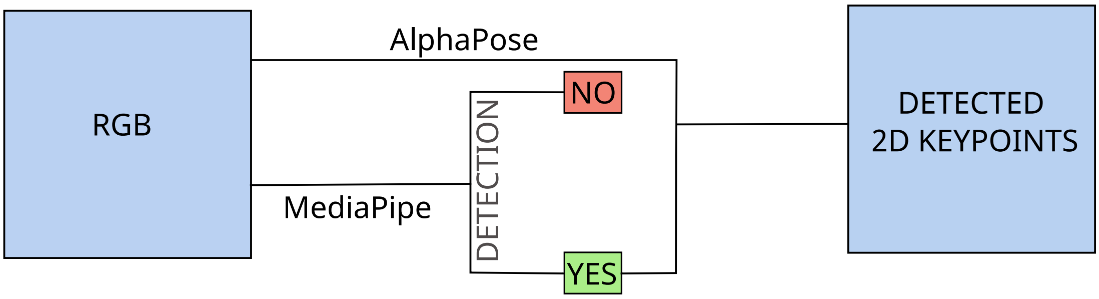


Figure 4.11: Human keypoints detection pipeline used for the 2D human keypoints detection in RGB images.

4.3 Relative Evaluation of the 3D Human Pose

According to Section 3.6.2 the evaluation of relative human body keypoints position has been done. In this experiment, the observed human body is static and the iCub robot observes the participant from 100 predefined positions. The goal of this evaluation is to determine which of the 2D to 3D transformation methods introduced in Section 3.5 provides more robust and accurate resulting 3D positions. The difference is in the method of processing the neighborhood of detected keypoint, namely the median or mean approach.

The first measurement was done on a participant sitting in front of the robot. The position of Intel RealSense was adjusted to 2 different positions with different field of view. The first position of the RealSense camera was heading directly forward with no tilt which results in observing mainly the upper body of the participant.

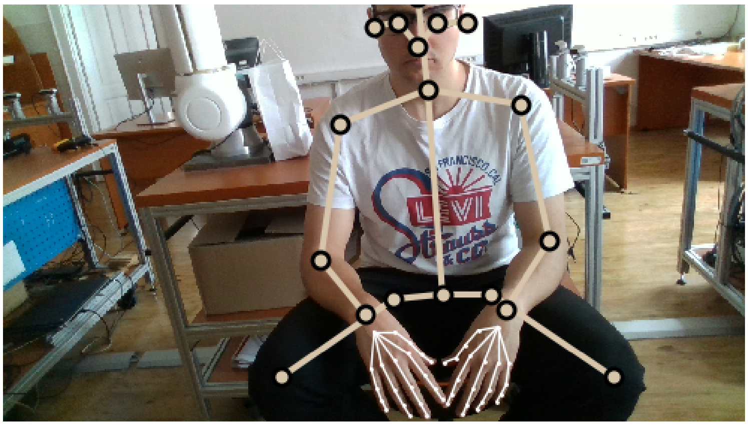
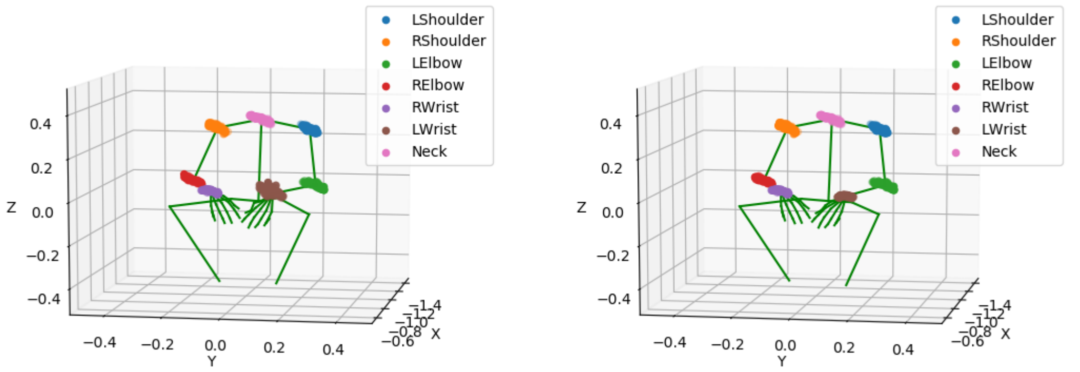


Figure 4.12: Example image during experiment with detected human body keypoints and RealSense position adjusted with no tilt.

We provide an example image in Figure 4.12 where the detection is performed as was proposed in Section 4.2 using AlphaPose wholebody keypoints detection in combination with MediaPipe hand detection. The keypoints detected in a 2D input image are transformed to 3D coordinates in the iCub *Root* frame following the process specified in Sections 3.4 and 3.5. For the 2D keypoint we also transform the neighborhood pixels and the resulting 3D position is computed either as a mean of the neighborhood points or as a median of these points.

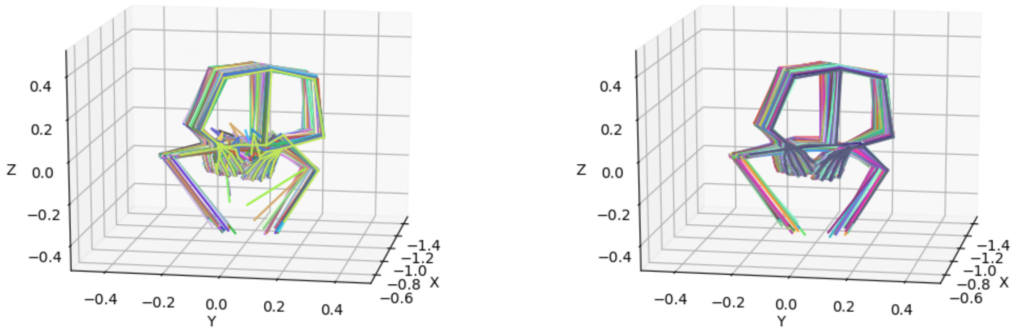


(a) 3D positions of selected keypoints computed using the mean method with the average human skeleton.

(b) 3D positions of selected keypoints computed using the median method with the average human skeleton.

Figure 4.13: Visualization of the 3D positions of detected keypoints and the human skeleton.

We processed the keypoints using both methods (mean and median) and visualized them in 3D space. Figure 4.13 shows the difference of 3D positions, where in Figure 4.13b the positions are more stable than in Figure 4.13a. The most visible discrepancy is in the positions of the left wrist keypoint. The detected human skeleton is an average over all skeletons detected in the 100 poses of the robot. We also provide a visualization of all the skeletons in Figure 4.14.



(a) 3D skeletons constructed from the keypoints using mean method.

(b) 3D skeletons constructed from the keypoints using median method.

Figure 4.14: Visualization of the 3D position of the human skeletons during the experiment.

There is a significant difference between the constructed skeletons as for the mean approach, Figure 4.14a, there are many wrong 3D keypoints positions in the hands area. The median approach, Figure 4.14b, provides more accurate and robust skeletons throughout the experiment. The important note is that the 2D detections in the images are the same for both used methods, and the differences are results of the different approaches only.

We also computed the standard deviation of distances from the average skeleton of the selected keypoints. These are listed in Table 4.10 in centimeters.

Approach	LWrist	RWrist	LElbow	RElbow	LShoulder	RShoulder
Median	0.8	0.9	1.1	0.9	0.8	1.1
Mean	2.1	0.9	1.1	1.3	0.8	1.2

Table 4.10: Standard deviations of distances from the mean human pose for both median and mean approaches. The deviation is in centimeters.

The resulting standard deviations correspond to the results of the observation of Figures 4.13 and 4.14. The biggest difference is again in the left wrist keypoint as the standard deviation is significantly higher for the mean method. The absolute values of the keypoints are appropriate to the rough estimation of the manually measured distances of the participant.

A similar human position was evaluated with tilting the Intel RealSense camera down. Due to tilt, the field of view focuses on the core and lower part of the human body, as shown in the example in Figure 4.15. We provide the same visualizations for this experiment setup, as well as the standard deviation statistics.

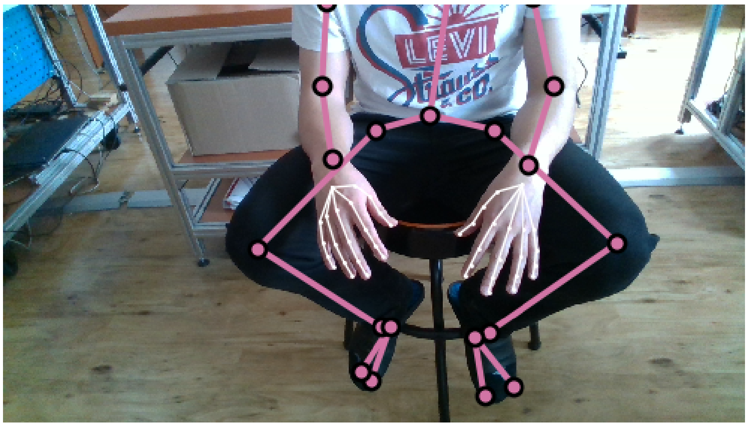
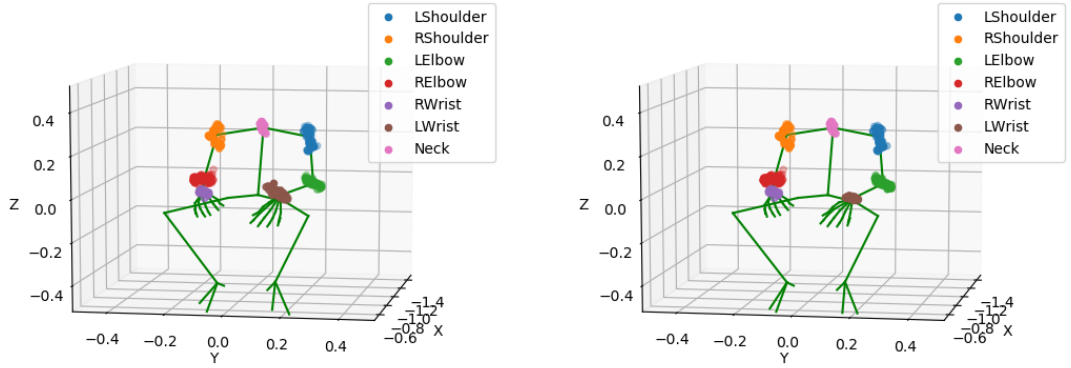


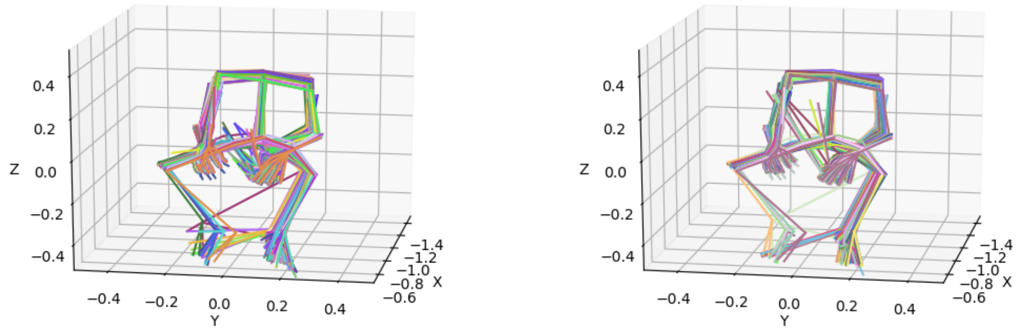
Figure 4.15: Example image during experiment with detected human body keypoints and RealSense position adjusted with tilt.



(a) 3D positions of selected keypoints computed using the mean method with the average human skeleton.

(b) 3D positions of selected keypoints computed using the median method with the average human skeleton.

Figure 4.16: Visualization of the 3D positions of detected keypoints and the human skeleton with tilted RealSense camera.



(a) 3D skeletons constructed from the keypoints using mean method.

(b) 3D skeletons constructed from the keypoints using median method.

Figure 4.17: Visualization of the 3D position of the human skeletons during the experiment with tilted RealSense camera.

Approach	LWrist	RWrist	LElbow	RElbow	LShoulder	RShoulder
Median	1.4	0.6	1.0	1.3	1.5	1.0
Mean	2.0	0.6	1.5	1.6	1.5	1.0

Table 4.11: Standard deviations of distances from the mean human pose for both median and mean approaches. The deviation is in centimeters.

The left shoulder keypoint is affected by the problematic 2D detection because the shoulders often are not visible and the estimations vary significantly. Also, the right elbow was affected by a less stable 2D keypoint detection in this setup. On the visualizations in Figures 4.16 and 4.17 the difference in both methods is visible, resulting in a similar conclusion as in the previous setup, where the median approach is more robust and provides more accurate 3D positions. The standard deviations are lower for the median approach and are listed in Table 4.11.

Summary

Resulting 3D positions of the detected human body keypoints are more accurate and robust using the median approach for computing 3D position from the detected keypoint neighborhood described in Section 3.5. Exact 3D positions also depend on the point of view of the camera, as the depth may vary due to different shape and size of each human body keypoint.

4.4 Absolute Evaluation of the 3D Human Pose

We prepared an experiment to compare the estimated (detected) 3D positions of keypoints to their ground truth positions obtained by Qualisys MoCap system as was stated in Section 3.6.2. Specifically, we marked the human body with 12 reflective markers on 6 keypoints, namely both shoulders, elbows and wrists, see Figure 4.18 for the demonstration. Each keypoint was marked with 2 markers to ensure its visibility by the MoCap system as well as more precise ground truth position estimation. The ground truth position of a keypoint was computed as a mean of both markers positions detected by the Qualisys MoCap. An extra marker was placed on the Intel RealSense camera to have an estimation of translation between the camera frame and the Qualisys reference frame.

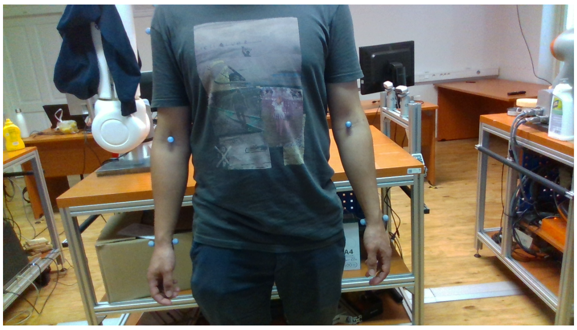
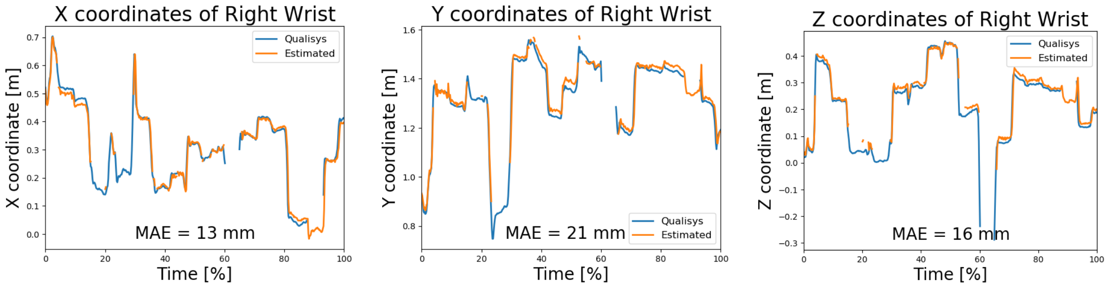


Figure 4.18: Image of the participant in the absolute evaluation of 3D human pose experiment with visible reflective markers placed on the body keypoints.

For computing the transformation between camera and Qualisys frame, we manually annotated 8 markers in 2 images from the Intel RealSense camera and computed the transformation using the SVD algorithm as described in Section 3.6.2. We checked that the translation part of the transformation corresponds to the marker located on the RealSense camera.

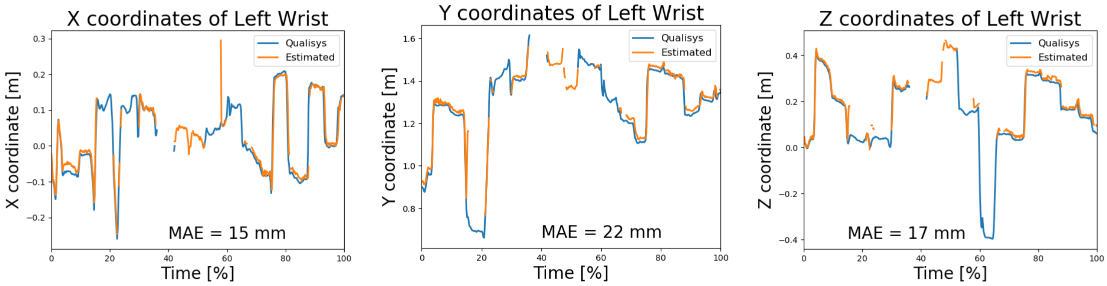
The human body keypoints were detected in RGB images using the AlphaPose detector and transformed from 2D to 3D using the method described in Section 3.5 with the median approach of the neighborhood of keypoint. At the 3D point we applied the transformation from the SVD algorithm, Equation 3.16.

The resulting positions of each marker keypoint were plotted in the following graphs alongside with the computed Mean Absolute Error (MAE) using Equation 3.17. The discontinuities in the graphs are caused by either not detecting the keypoint in the RGB image from the RealSense camera as they did not occur in the images or the marker points were not visible in the MoCap system.



(a) (b) (c)

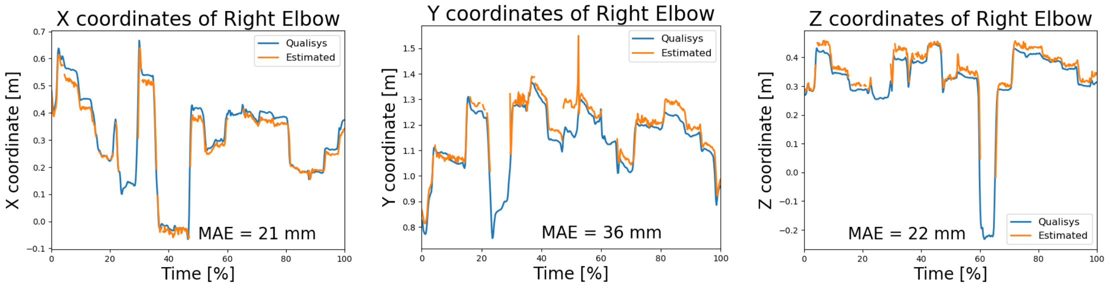
Figure 4.19: Graphs with right wrist coordinates during the experiment and Mean Absolute Errors (MAE). Keypoint detected using AlphaPose and MediaPipe hand detector and transformed to 3D position with computing the median of the points in the keypoint neighborhood.



(a) (b) (c)

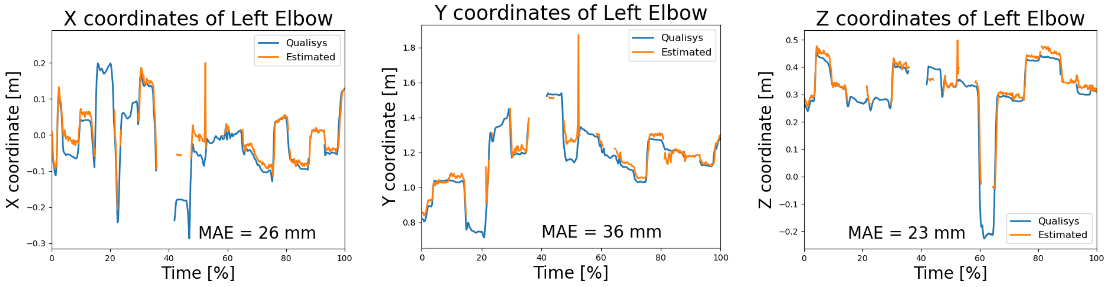
Figure 4.20: Graphs with left wrist coordinates during human motion and Mean Absolute Errors (MAE). Keypoint detected using AlphaPose and MediaPipe hand detector and transformed to 3D position with computing the median of the points in the keypoint neighborhood.

Figures 4.19 and 4.20 show that wrists were detected correctly and the 3D positions more or less align with the ground truth from the Qualisys MoCap. The Mean Absolute Errors are around 1-2 cm, which is caused mainly due to imprecise 2D detections meaning that the human keypoints were detected at a slightly different position in comparison with the marked positions. On the other hand, 2 centimeters are in bounds of the size of human wrist.



(a) **(b)** **(c)**

Figure 4.21: Graphs with right elbow coordinates during human motion and Mean Absolute Errors (MAE). Keypoint detected using AlphaPose and MediaPipe hand detector and transformed to 3D position with computing the median of the points in the keypoint neighborhood.



(a) **(b)** **(c)**

Figure 4.22: Graphs with left elbow coordinates during human motion and Mean Absolute Errors (MAE). Keypoint detected using AlphaPose and MediaPipe hand detector and transformed to 3D position with computing the median of the points in the keypoint neighborhood.

The 3D positions of both elbows have slightly higher MAEs, 2-4 cm, as visualized in Figures 4.21 and 4.22. The reason is the same as for the wrists, as the 2D detections do not correspond precisely with the marked positions. Moreover, the elbow is a bigger joint than the wrist, so the errors are greater. There is also a spike for both elbows in the middle of the experiment, where both elbows were overlaid with the wrists in the RGB image and the depth from the RealSense camera was not correct for the elbow keypoints, as can be seen in Figure 4.23.

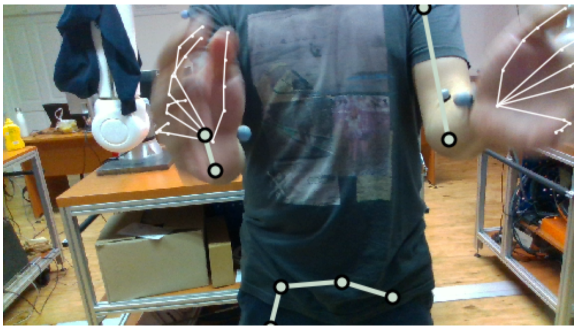
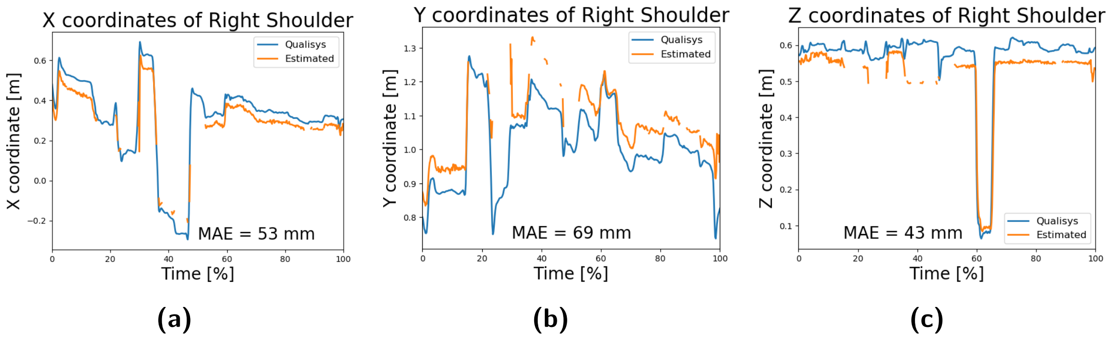
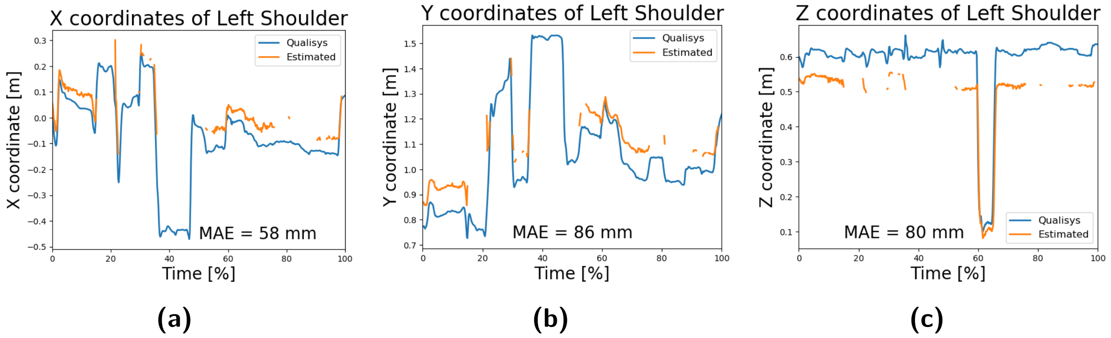


Figure 4.23: Image causing the spike in the 3D positions of both elbows.



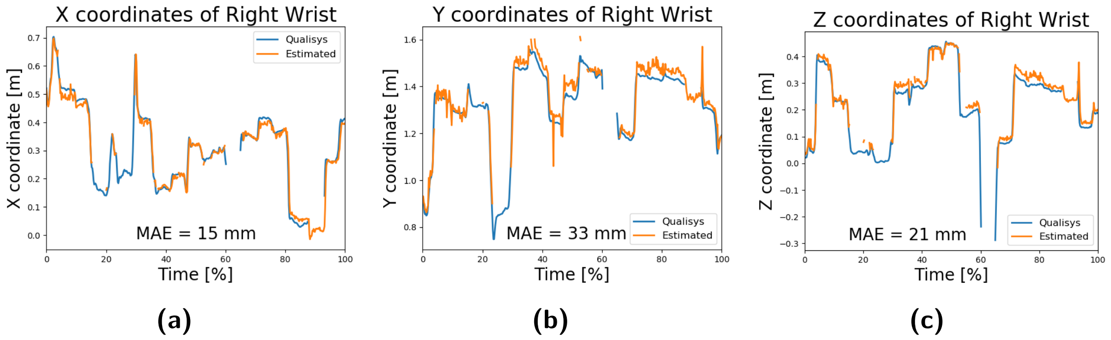
(a) (b) (c)

Figure 4.24: Graphs with right shoulder coordinates during human motion and Mean Absolute Errors (MAE). Keypoint detected using AlphaPose and MediaPipe hand detector and transformed to 3D position with computing the median of the points in the keypoint neighborhood.



(a) (b) (c)

Figure 4.25: Graphs with left shoulder coordinates during human motion and Mean Absolute Errors (MAE). Keypoint detected using AlphaPose and MediaPipe hand detector and transformed to 3D position with computing the median of the points in the keypoint neighborhood.



(a) (b) (c)

Figure 4.26: Graphs with right wrist coordinates during human motion and Mean Absolute Errors (MAE). Keypoint detected using AlphaPose and MediaPipe hand detector and transformed to 3D position with computing the mean of the points in the keypoint neighborhood.

The shoulder 3D positions during the experiment are visualized in Figures 4.24 and 4.25. The graphs and also the Mean Absolute Errors are worse in comparison with other keypoints. The error caused by the 2D detection is more significant for the shoulder detection because it is the biggest joint and the marker points were placed on the outter part of the shoulder joint whereas the detection usually detects the shoulder in the inner part of the joint.

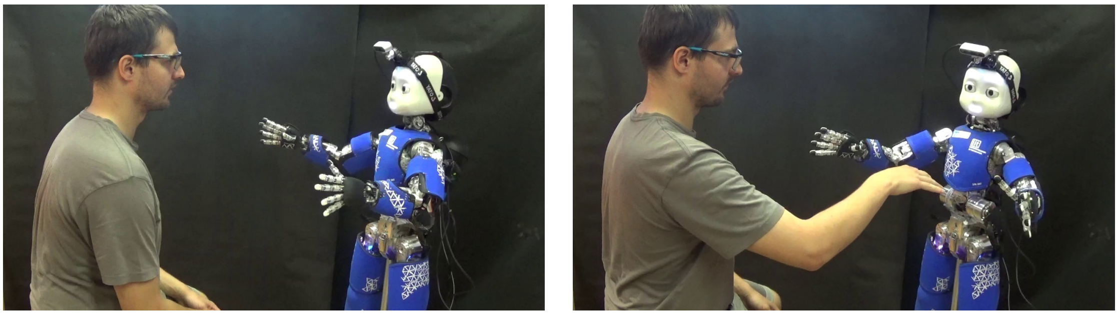
We also provide the graph with the MAE of the right wrist when the 3D position is determined from the keypoint neighborhood using the mean approach. We can compare Figures 4.19 and 4.26. The resulting MAEs are worse by about 1 cm for the mean approach, and the detected coordinates are noisier and less accurate. This tendency is also present for the other keypoints detected in this experiment, but for simplicity, we did not provide the resulting graphs and MAEs.

Summary

The 3D positions depend heavily on the correct 2D detection in the input image and the clearance of visibility of the keypoint in order to be able to determine the correct depth of the keypoint. If the keypoint is detected and visible flawlessly, then the 3D position is computed correctly with the error in the bounds of the size of keypoint. As well as in the results of experiment in Section 4.3 the median method for extracting the position from the keypoint neighborhood is more accurate and robust.

4.5 Demonstration on Human-Robot Scenario

As a demonstration of the proposed solution, we prepared a human-robot interaction scenario, when the iCub robot holds the defined position of the left arm. When the participant tries to reach the defined hold position of the left arm of the iCub, the robot should avoid him and move away as shown in Figure 4.27. The holding position of the iCub robot is visualized in Figure 4.27a and the avoidance maneuver is visible in Figure 4.27b. The control module used for the avoidance maneuver was adjusted and prepared by Ing. Jakub Rozlivek. A video of the demonstration experiment is available at `https://www.youtube.com/watch?v=z-mdR8C6o9g`.



(a) iCub robot holding position of the left arm.

(b) iCub robot with the moved left arm because of the human body avoidance.

Figure 4.27: Demonstration of the human body avoidance scenario.

During the experiment, the participant was instructed to perform 3 different poses. The first pose was sitting in front of the robot in its field of view and reaching the critical position with the participants hand. The second pose of the participant included standing in front of the robot and reaching the position with the hand again. In the last pose, the participant tries to displace the robot using his head. The described poses of the participant will be shown in the following figures.

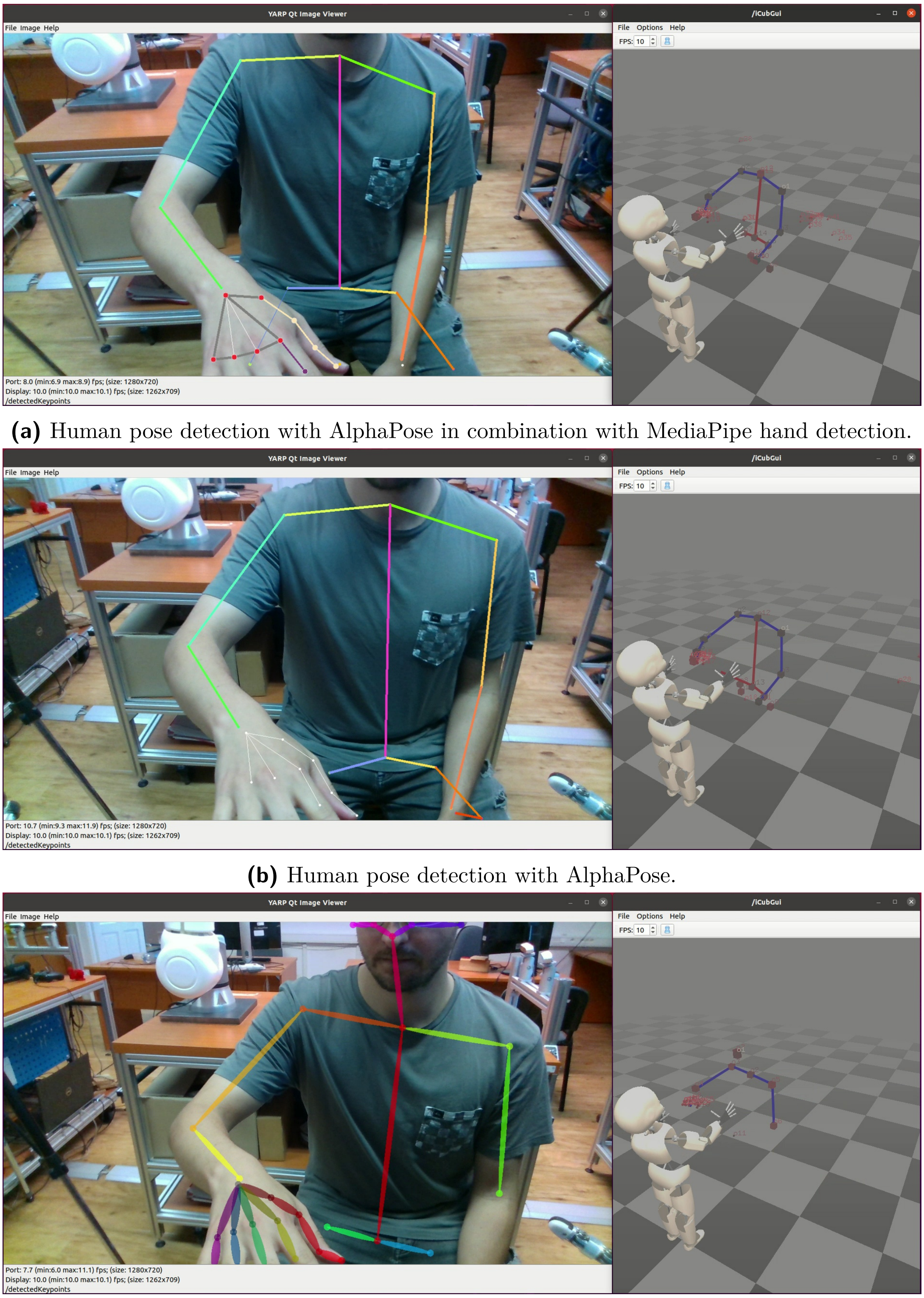
The proposed solution consists of the human keypoints detection using AlphaPose and MediaPipe hand detectors with transformation to 3D positions using the median approach for processing the keypoint neighborhood. For comparison we also executed this experiment with the solo AlphaPose human keypoints detection as well as the detection using only OpenPose, which is used in the `yarpOpenPose` iCub module. We will compare these different detection approaches in each of the participant poses.

Firstly, we will compare the different detection methods on the first pose with the participant sitting in front of the robot and reaching its left arm. As can be seen in Figure 4.28, all of the detectors are able to detect the human body keypoints correctly alongside with the correct 3D positions. The correct detection caused the iCub robot to move his left arm to avoid the participants hand.

The second comparison of the detectors for the demonstration includes the standing pose of the participant visualized in Figure 4.29. Whereas the AlphaPose is able to detect the human in the standing position in Figure 4.29b, OpenPose, Figure 4.29c, not only does not detect the finger keypoints, but it is unable to detect the wrist or the elbow keypoint even with having subjectively the best view of the human body. As a result, in the OpenPose experiment, the robot did not move his left arm. As we stated earlier, MediaPipe provides more precise finger keypoints detection, as can be seen in Figure 4.29a. In both cases, AlphaPose with MediaPipe detection and AlphaPose only, the robot moved in order to let the participant reach the crucial position.

As the last compared position, the participant tried to move the robot using his head. In this case, AlphaPose again overperformed the OpenPose detection, illustrated in Figure 4.30, as AlphaPose was able to detect the human body unlike OpenPose. In this participants position setup, the hand is not occuring in the RGB image and thus the MediaPipe does not provide any benefits.

The conclusion of the demonstration is that the AlphaPose proved to be the best human body keypoint detector and is able to perform even better when used alongside MediaPipe hand detection, as it provides more accurate finger keypoints detection. In terms of the 3D position computation, the results are accurate enough to be able to localize the detected human keypoints in order to control the robot avoidance.



```
Port: 8.0 (min:6.9 max:8.9) fps; (size: 1280x720)
Display: 10.0 (min:10.0 max:10.1) fps; (size: 1262x709)
/detectedKeypoints
```

(a) Human pose detection with AlphaPose in combination with MediaPipe hand detection.

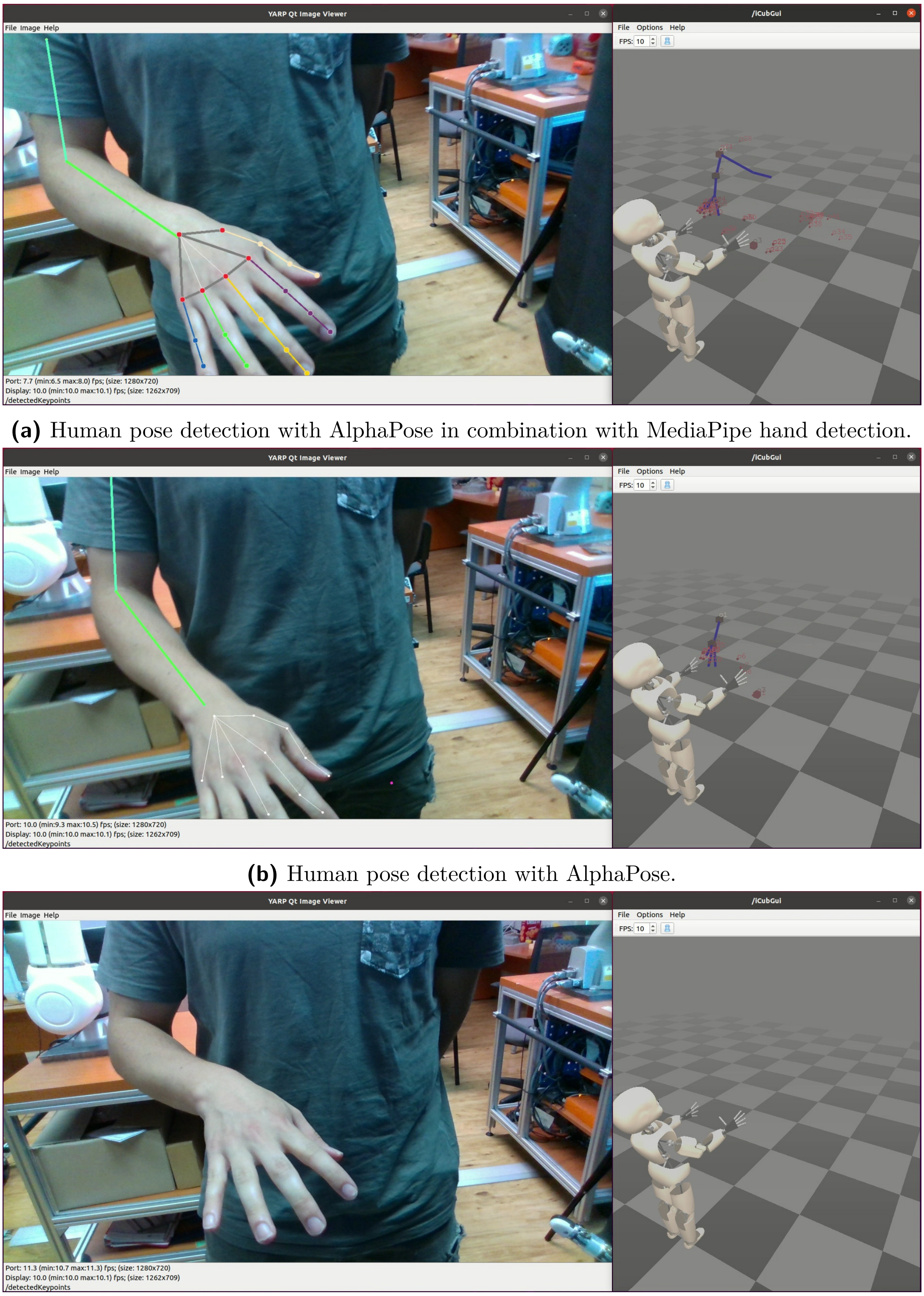
```
Port: 10.7 (min:9.3 max:11.9) fps; (size: 1280x720)
Display: 10.0 (min:10.0 max:10.1) fps; (size: 1262x709)
/detectedKeypoints
```

(b) Human pose detection with AlphaPose.

```
Port: 7.7 (min:6.0 max:11.1) fps; (size: 1280x720)
Display: 10.0 (min:10.0 max:10.1) fps; (size: 1262x709)
/detectedKeypoints
```

(c) Human pose detection with OpenPose

Figure 4.28: Comparison of the human body keypoints detections in the sitting pose of the participant reaching the iCub left arm. Keypoints detection is visualized in the left part of images; 3D positions of the keypoints visualized in the right part of the images



```
Port: 7.7 (min:6.5 max:8.0) fps; (size: 1280x720)
Display: 10.0 (min:10.0 max:10.1) fps; (size: 1262x709)
/detectedKeypoints
```

(a) Human pose detection with AlphaPose in combination with MediaPipe hand detection.

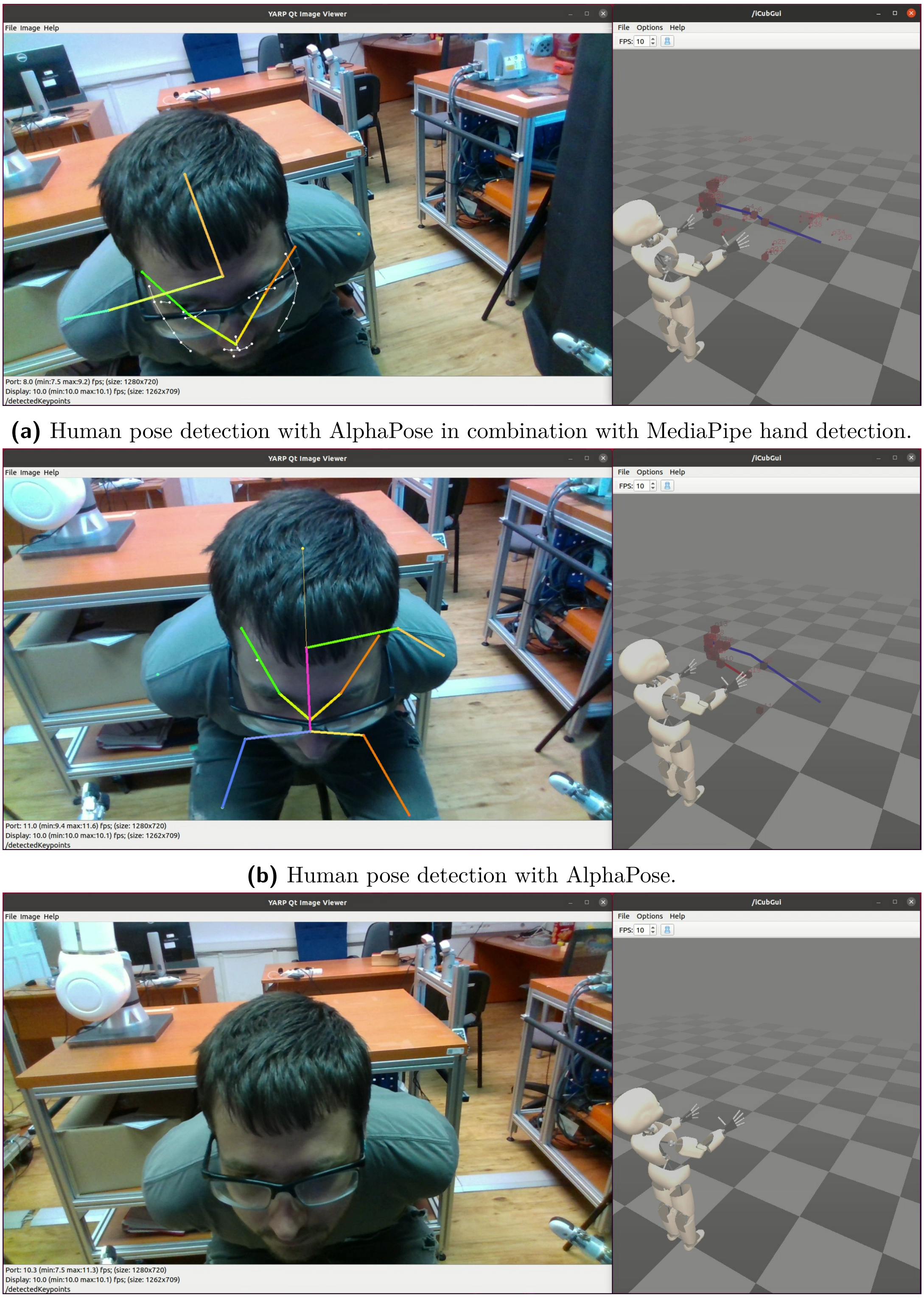
```
Port: 10.0 (min:9.3 max:10.5) fps; (size: 1280x720)
Display: 10.0 (min:10.0 max:10.1) fps; (size: 1262x709)
/detectedKeypoints
```

(b) Human pose detection with AlphaPose.

```
Port: 11.3 (min:10.7 max:11.3) fps; (size: 1280x720)
Display: 10.0 (min:10.0 max:10.1) fps; (size: 1262x709)
/detectedKeypoints
```

(c) Human pose detection with OpenPose

Figure 4.29: Comparison of the human body keypoints detections in the standing pose of the participant reaching the iCub left arm. Keypoints detection is visualized in the left part of images; 3D positions of the keypoints visualized in the right part of the images



```
Port: 8.0 (min:7.5 max:9.2) fps; (size: 1280x720)
Display: 10.0 (min:10.0 max:10.1) fps; (size: 1262x709)
/detectedKeypoints
```

(a) Human pose detection with AlphaPose in combination with MediaPipe hand detection.

```
Port: 11.0 (min:9.4 max:11.6) fps; (size: 1280x720)
Display: 10.0 (min:10.0 max:10.1) fps; (size: 1262x709)
/detectedKeypoints
```

(b) Human pose detection with AlphaPose.

```
Port: 10.3 (min:7.5 max:11.3) fps; (size: 1280x720)
Display: 10.0 (min:10.0 max:10.1) fps; (size: 1262x709)
/detectedKeypoints
```

(c) Human pose detection with OpenPose

Figure 4.30: Comparison of the human body keypoints detections in the scenario with the participant reaching the iCub left arm with his head. Keypoints detection is visualized in the left part of images; 3D positions of the keypoints visualized in the right part of the images

Chapter 5

Conclusion, Discussion and Future Work

In this chapter, we will summarize the results and methods in Section 5.1. The drawbacks and limitations of the proposed solution will be discussed in Section 5.2. In Section 5.3 we will present possible improvements for future work.

5.1 Conclusion

In this thesis, we created the Close Proximity Dataset and the Close Proximity Dataset with Excluded Head, Section 3.2, from the existing Halpe validation dataset, including people in simulated close proximity with their wholebody keypoint annotations. Datasets available at [34]. Both datasets were created by cropping the original images according to people occurring in the images, resulting in the Close Proximity Dataset with 1624 annotated images and the Close Proximity Dataset with Excluded Head with 1608 annotated images.

These datasets were used for the evaluation and comparison of selected state-of-the-art 2D human body keypoint detectors (listed in Section 3.3) in Section 4.1. As a result, the best performing detectors were AlphaPose and MMPose human body keypoint detectors and the MediaPipe detector for the finger keypoints detection. These were qualitatively evaluated in Section 4.2 and the AlphaPose human pose detector and the MediaPipe hand detector were selected as the best performing in terms of the keypoint detection accuracy and robustness of keypoint detection, as well as the speed of detection. Furthermore, we proposed a combined solution for the detection of the 2D human keypoints using simultaneous detection by AlphaPose in combination with MediaPipe for the finger keypoints. The schematic visualization is available in Figure 4.11, where the MediaPipe finger keypoints detection is preferred over the finger keypoints detected by AlphaPose. This solution provides even more robust 2D detection of finger keypoints, which are one of the most important to detect for human-robot interaction (HRI).

A method for obtaining the 3D positions of the detected keypoints was proposed in Section 3.5 taking into account a 2D neighborhood of the keypoint. The size of the neighborhood is determined dynamically based on the distance from the camera, as well as the type of the keypoint. The points in the neighborhood of the keypoint were transformed to the 3D camera coordinate frame using Equations 3.3, 3.4 and 3.5. Additionally, we proposed 2 approaches for determining the resulting 3D position computing either the (i) mean or the (ii) median of the 3D positions in the neighborhood. Finally, the 3D positions were transformed to the iCub *Root* coordinate frame as described in Section 3.4.

Firstly, the proposed method for the computation of the 3D keypoint positions was evaluated in Section 4.3 using the relative evaluation with a static human participant that was supposed to be detected by the moving robot. The conclusion of this experiment was that the 3D positions depend heavily on the correct 2D keypoints detection, as expected. However, the median approach for obtaining the final 3D position from the neighborhood of the keypoint was shown to be more robust and accurate than the mean approach.

Secondly, the 3D keypoint positions were evaluated in Section 4.4 using the absolute evaluation as well. The resulting 3D positions were compared with respect to the estimated ground truth positions provided by the Qualisys Motion Capture (MoCap) system. The 3D coordinates of the keypoints corresponded to the ground truth positions with Mean Absolute Errors (MAEs) in bounds of the keypoint (joint) sizes. In addition, we provided a comparison for the mean and the median approaches as well, with the same results, as the median approach is more accurate and robust.

Finally, the proposed pipeline solution consisting of the 2D human keypoints detection using AlphaPose and MediaPipe detectors followed by the 2D to 3D transformation with the median approach was demonstrated in Section 4.5 in a HRI scenario with the iCub humanoid robot equipped with the Intel RealSense camera. Furthermore, we compared other 2D detection methods in the human-robot interaction scenario, with the conclusion that the proposed combined 2D detection solution provides the most robust and accurate results. The accompanying video is available at `https://www.youtube.com/watch?v=z-mdR8C6o9g`.

5.2 Discussion

Our method for close proximity keypoint detection does not deal with a partially occluded human body ideally. Although we detect occluded 2D keypoints, the depth information for them is not correct due to occlusion. In our setup with a camera located on the robot and providing first person POV, this issue does not directly imply problems for the safety as it only results in 3D positions closer to the robot. However, the detected human pose is not detected correctly, e.g. for human action recognition or gesture recognition.

Another drawback of the proposed solution is the Intel RealSense camera depth framerate. As we stated earlier in Section 4.2, the 2D keypoint detection runs at approximately 15 fps. But the framerate of the Intel RealSense included in the YARP system is only 8 fps. As a result, the framerate of the depth information becomes a bottleneck in the processing pipeline. According to documentation, the Intel RealSense camera should be able to provide depth information even with up to 90 fps. A possible solution to this bottleneck would be to improve the implementation of the Intel RealSense device in the YARP system.

During the development of the solution, we encountered a few limitations. The first limitation were the datasets. Most of the dataset images do not contain people in close proximity with their wholebody annotation. For this purpose, we created the Close Proximity Datasets with cropping the images which lead to lower resolution images. However, the resulting lower resolution influences the performance of the detectors mainly for the finger keypoints detection.

Another limitation were different outputs of the detectors, which led to the necessity of different postprocessing of the detectors outputs. Even though this limitation was solved, it required additional time dedicated to this issue. Moreover, to change the detection method in the processing pipeline, it is required to treat the whole 2D detection pipeline individually, which is not development-friendly.

5.3 Future Work

The close proximity human detection is a complex task, and there are multiple ways how to improve our presented solution. For the purposes of even more accurate and robust 2D detection, the real dataset with people in the close proximity could be created alongside with a precise wholebody annotation, ideally manually annotated. Thanks to such dataset, it would be possible to finetune the AlphaPose (or others as well) neural network in order to perform better on people in the close proximity.

Another improvement would be to solve the issue with the partial occlusion of the human body. It would be possible to take into account the *SMPL* [17] or *SMPL-X* [20] models in order to correct the 3D keypoint positions accordingly. Or even try the 3D pose detection in the RGB images directly.

For a better overall performance of human detection, the RGB-D sensor could be improved as well. As we stated earlier, we could improve the Intel RealSense performance in the YARP system. Thanks to that, the human body detection would be even more safe and robust for the HRI scenarios thanks to the more frequent updates of the human pose.

Bibliography

[1] G. Metta, L. Natale, F. Nori, G. Sandini, D. Vernon, L. Fadiga, C. von Hofsten, K. Rosander, M. Lopes, J. Santos-Victor, A. Bernardino, and L. Montesano, “The iCub humanoid robot: An open-systems platform for research in cognitive development,” *Neural Networks*, vol. 23, no. 8, pp. 1125–1134, 2010, social Cognition: From Babies to Robots.

[2] P. Svarny, M. Tesar, J. K. Behrens, and M. Hoffmann, “Safe physical HRI: Toward a unified treatment of speed and separation monitoring together with power and force limiting,” in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 7580–7587.

[3] G. Metta, P. Fitzpatrick, and L. Natale, “YARP: Yet Another Robot Platform,” *International Journal of Advanced Robotic Systems*, vol. 3, no. 1, p. 8, 2006.

[4] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh, “OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[5] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, “Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7291–7299.

[6] T. Simon, H. Joo, I. Matthews, and Y. Sheikh, “Hand Keypoint Detection in Single Images using Multiview Bootstrapping,” in *Conference on Computer Vision and Pattern Recognition*, 2017.

[7] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, “Convolutional Pose Machines,” in *Conference on Computer Vision and Pattern Recognition*, 2016.

[8] MMPose Contributors, “OpenMMLab Pose Estimation Toolbox and Benchmark,” https://github.com/open-mmlab/mmpose, 2020.

[9] H.-S. Fang, S. Xie, Y.-W. Tai, and C. Lu, “RMPE: Regional Multi-person Pose Estimation,” in *International Conference on Computer Vision*, 2017.

[10] J. Li, C. Wang, H. Zhu, Y. Mao, H.-S. Fang, and C. Lu, “Crowdpose: Efficient Crowded Scenes Pose Estimation and A New Benchmark,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 10 863–10 872.

[11] Y. Xiu, J. Li, H. Wang, Y. Fang, and C. Lu, "Pose Flow: Efficient Online Pose Tracking," in *British Machine Vision Conference*, 2018.

[12] Y. Cheng, B. Yang, B. Wang, Y. Wending, and R. Tan, "Occlusion-Aware Networks for 3D Human Pose Estimation in Video," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 723–732.

[13] B. Rim, N.-J. Sung, J. Ma, Y.-J. Choi, and M. Hong, "Real-time Human Pose Estimation using RGB-D images and Deep Learning," *Journal of Internet Computing and Services*, vol. 21, no. 3, pp. 113–121, 2020.

[14] C. Zimmermann, T. Welschehold, C. Dornhege, W. Burgard, and T. Brox, "3D Human Pose Estimation in RGBD Images for Robotic Task Learning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1986–1992.

[15] K. Buys, C. Cagniart, A. Baksheev, T. De Laet, J. De Schutter, and C. Pantofaru, "An adaptable system for RGB-D based human body detection and pose estimation," *Journal of Visual Communication and Image Representation*, vol. 25, no. 1, pp. 39–52, 2014, visual Understanding and Applications with RGB-D Cameras.

[16] D. H. P. Nguyen, M. Hoffmann, A. Roncone, U. Pattacini, and G. Metta, "Compact Real-Time Avoidance on a Humanoid Robot for Human-Robot Interaction," in *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, ser. HRI '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 416–424.

[17] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "SMPL: A Skinned Multi-Person Linear Model," *ACM transactions on graphics (TOG)*, vol. 34, no. 6, pp. 1–16, 2015.

[18] F. Bogo, A. Kanazawa, C. Lassner, P. Gehler, J. Romero, and M. J. Black, "Keep it SMPL: Automatic Estimation of 3D Human Pose and Shape from a Single Image," in *European conference on computer vision*. Springer, 2016, pp. 561–578.

[19] A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik, "End-to-end Recovery of Human Shape and Pose," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7122–7131.

[20] G. Pavlakos, V. Choutas, N. Ghorbani, T. Bolkart, A. A. Osman, D. Tzionas, and M. J. Black, "Expressive Body Capture: 3D Hands, Face, and Body from a Single Image," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 10 975–10 985.

[21] M. Omran, C. Lassner, G. Pons-Moll, P. Gehler, and B. Schiele, "Neural Body Fitting: Unifying Deep Learning and Model-Based Human Pose and Shape Estimation," in *2018 international conference on 3D vision (3DV)*. IEEE, 2018, pp. 484–494.

[22] G. Pavlakos, L. Zhu, X. Zhou, and K. Daniilidis, "Learning to Estimate 3D Human Pose and Shape from a Single Color Image," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 459–468.

[23] Y. Rong, T. Shiratori, and H. Joo, “FrankMocap: A Monocular 3D Whole-Body Pose Estimation System via Regression and Integration,” in *IEEE International Conference on Computer Vision Workshops*, 2021.

[24] H. Joo, N. Neverova, and A. Vedaldi, “Exemplar Fine-Tuning for 3D Human Pose Fitting Towards In-the-Wild 3D Human Pose Estimation,” *International Conference on 3D Vision*, 2021.

[25] R. A. Güler, N. Neverova, and I. Kokkinos, “Densepose: Dense Human Pose Estimation in the Wild,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7297–7306.

[26] G. Varol, J. Romero, X. Martin, N. Mahmood, M. J. Black, I. Laptev, and C. Schmid, “Learning from Synthetic Humans,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 109–117.

[27] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft COCO: Common Objects in Context,” in *European conference on computer vision.* Springer, 2014, pp. 740–755.

[28] S. Jin, L. Xu, J. Xu, C. Wang, W. Liu, C. Qian, W. Ouyang, and P. Luo, “Whole-Body Human Pose Estimation in the Wild,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[29] Y.-L. Li, L. Xu, X. Liu, X. Huang, Y. Xu, S. Wang, H.-S. Fang, Z. Ma, M. Chen, and C. Lu, “PaStaNet: Toward Human Activity Knowledge Engine,” in *Conference on Computer Vision and Pattern Recognition*, 2020.

[30] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung, “A Benchmark Dataset and Evaluation Methodology for Video Object Segmentation,” in *Computer Vision and Pattern Recognition*, 2016.

[31] M. Andriluka, U. Iqbal, E. Ensafutdinov, L. Pishchulin, A. Milan, J. Gall, and S. B., “PoseTrack: A Benchmark for Human Pose Estimation and Tracking,” in *Conference on Computer Vision and Pattern Recognition*, 2018.

[32] D. Damen, H. Doughty, G. M. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, and M. Wray, “The EPIC-KITCHENS Dataset: Collection, Challenges and Baselines,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 43, no. 11, pp. 4125–4141, 2021.

[33] D. Damen, H. Doughty, G. M. Farinella, , A. Furnari, J. Ma, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, and M. Wray, “Rescaling Egocentric Vision: Collection, Pipeline and Challenges for EPIC-KITCHENS-100,” *International Journal of Computer Vision (IJCV)*, 2021.

[34] J. Dočekal and M. Hoffmann, 2022. [Online]. Available: https://osf.io/qfkvt/?view_only=c62a123dc78c4cc7a8504eb36da04b2f

[35] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, “2D Human Pose Estimation: New Benchmark and State of the Art Analysis,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[36] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2," https://github.com/facebookresearch/detectron2, 2019.

[37] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M. Yong, J. Lee, W.-T. Chang, W. Hua, M. Georg, and M. Grundmann, "MediaPipe: A Framework for Perceiving and Processing Reality," in *Third Workshop on Computer Vision for AR/VR at IEEE Computer Vision and Pattern Recognition (CVPR) 2019*, 2019.

[38] A. Vakunov, C.-L. Chang, F. Zhang, G. Sung, M. Grundmann, and V. Bazarevsky, "MediaPipe Hands: On-device Real-time Hand Tracking," 2020.

[39] F. Zhang, V. Bazarevsky, A. Vakunov, A. Tkachenka, G. Sung, C.-L. Chang, and M. Grundmann, "MediaPipe Hands: On-device Real-time Hand Tracking," 2020. [Online]. Available: https://arxiv.org/abs/2006.10214

[40] M. Ruggero Ronchi and P. Perona, "Benchmarking and Error Diagnosis in Multi-Instance Pose Estimation," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 369–378.

[41] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-Squares Fitting of Two 3-D Point Sets," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, no. 5, pp. 698–700, 1987.