Towards Real-time Physical Human-Robot Interaction using Skeleton Information and Hand Gestures

Osama Mazhar Sofiane Ramdani Benjamin Navarro Robin Passama Andrea Cherubini

***Abstract*— For successful physical human-robot interaction, the capability of a robot to understand its environment is imperative. More importantly, the robot should extract from the human operator as much information as possible. A reliable 3D skeleton extraction is essential for a robot to predict the intentions of the operator while he/she moves toward the robot or performs a gesture with a specific meaning. For this purpose, we have integrated a time-of-flight depth camera with a state-of-the-art 2D skeleton extraction library namely *Openpose*, to obtain 3D skeletal joint coordinates reliably. We have also developed a robust and rotation invariant (in the coronal plane) hand gesture detector using a convolutional neural network. At run time (after having been trained) the detector does not require any pre-processing of the hand images. A complete pipeline for skeleton extraction and hand gesture recognition is developed and employed for real-time physical human-robot interaction, demonstrating the promising capability of the designed framework. This work establishes a firm basis and will be extended for the development of intelligent human intention detection in physical human-robot interaction scenarios, to efficiently recognize a variety of static as well as dynamic gestures.**

I. INTRODUCTION

The recent development of light-weight robot manipulators and integration of mobile robots in both industrial and service applications has triggered attention on the research of safe physical human-robot interaction (pHRI). The appropriate understanding of the user, his/her safety, reliable performance in varying environments and real-time operation are all key-factors in pHRI studies. Advances in computer hardware, vision sensors and software have enabled robots to become more useful in their work environment. In particular, depth cameras like Microsoft Kinect, Orbbec Astra and Intel SR300 are becoming increasingly popular among computer-vision and robotic researchers for the development of robust pHRI applications.

A well known study [1] shows that 93% of the human communication is non-verbal and 55% of this is accounted for elements like facial expressions, posture, etc. In this perspective, capabilities like gesture recognition and human behavior understanding may be extremely useful for a robotic system in pHRI scenarios [2]. Gesture recognition is an active field of research in computer vision and is an effective way of communicating with a robot [3]. In this paper, we propose a pHRI framework which enables a robot to understand and to obey the commands given by the human-operator in the form of hand gestures.

All authors are with CNRS-University of Montpellier, LIRMM, Interactive Digital Humans group, 161 rue Ada, 34095, Montpellier, France osama.mazhar@lirmm.fr

Background and related work are described in Sect. II. In Sect. III, our contributions are stated briefly. Methodology is detailed in Sect. IV while a pHRI experiment and results are explained in Sect. V. Section VI concludes the paper.

II. BACKGROUND AND RELATED WORK

In [4], the authors have classified gestures in three types:

1) hand and arm gestures,
2) body gestures: full body motions and gait,
3) head and face gestures: nodding or shaking head, direction of eye-gaze, expressions of emotions.

According to [3], the essential parts of gesture recognition are sensor data collection, hand/body localization, feature tracking for dynamic gestures and gesture classification.

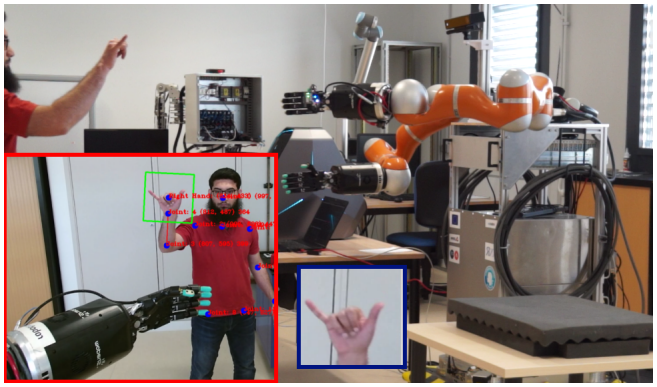

Fig. 1. Overall proposed pHRI scenario with BAZAR - The dual-arm mobile manipulator developed at LIRMM.

In [2], an image based human-robot collaboration system is proposed using a Kinect mounted on a wifibot which carries a NAO robot. The robot is able to navigate towards an object pointed on the floor. The proposed facial tracker fails to detect gestures quite often, as untrained users make those gestures subtly. Besides, no physical interaction between the human and user is present in this work. A similar – but hardware demanding – scenario is proposed in [5] where three Kinects are mounted around the workspace of a mobile robot. The authors detect dynamic gestures using a Fast-Fourier Transform which is used to segment a gesture, through the estimation of its period. This requires continuous repetition of each gesture in a loop several times in pre-operational/training phases. Training neural networks only on pre-processed images can prevent them from extracting and learning diverse features and may compromise the detector performance during recognition phase. The authors train 10

different neural networks for each gesture and this requires substantial resources. Moreover, this scenario involves no physical interaction between robot and human. In [6], Kinect-based object recognition through 3D gestures is proposed. The OpenNI and NITE middleware are used to extract the skeleton information of the human standing in front of the camera. The object location is fixed and a rigid object segmentation procedure is used with predefined constraints (e.g., the table color is white). Such conditions may not always be present in a real human-robot interaction task. Also, the removal of background using depth information may fail in some conditions (e.g., when the human operator stands near a wall). The objects chosen in the demonstration appear to be only of rectangular/box shape thus are detected using corner detectors. The histogram matching algorithm is used to recognize the objects.

This has been outperformed by modern deep learning techniques like Convolutional Neural Networks (CNN) [7]. Recently, [8] makes use of CNN for hand gesture recognition for a Human-Computer Interface. The author proposes a color independent classifier by feeding a pre-processed binary images into a LeNet network [9]. This makes classification accuracy dependent on the pre-processing step although, if provided with sufficient data, CNN are inherently robust enough to learn color features. In [10], the authors propose a HRI system for navigation of a mobile robot using a Kinect. For body and hand skeleton detection, a skeleton topology with multiple nodes is fit on the point cloud acquired from the sensor. This technique is not reliable, as both skeleton and hands have several non-linear anatomical constraints, that make the task of accurate pose detection difficult. A system targeting pHRI is proposed in [11], where a human-user gives commands to a robotic arm to follow, grasp, move and place an object. The arm gestures are used to control the robot, so that the gestures are distinguished with respect to predefined elbow angle ranges. The method does not incorporate hand gestures detection. This also makes the interaction system less intuitive for comfortable human-robot interaction tasks, as the human operator will have to learn the required elbow angles. Besides, a color-coordinate based algorithm is proposed for object detection, limiting the detection of multi-color objects. In [12], the author uses the skin color for hand segmentation assuming a planar background. Although the skeleton of the hand is extracted using distance transform, the approach only works with open hand gestures and mostly when the palm is facing the camera. The authors of [13] propose a technique to navigate a mobile robot with a Kinect. The OpenNI middleware is used to extract hand position and no physical interaction is present.

The localization of human body and of its sub-parts (e.g., hands or face) depends on the choice of sensor used and on its output. In [8] and [12], the authors use hand color filtering to localize hands in the scene. In [13], the human is detected by a laser sensor and hands are localized using OpenNI as in [5]. The authors of [10] localize the body using a technique inspired by [14] to merge clusters of a point cloud from Kinect after voxel filtering and ground plane removal. Yang et al. [11] use Microsoft SDK; object searching is done on the basis of color and shape of the object point cloud. Tracking of hands and fingers can also be done by optical and infrared based sensors like *Leap Motion*. This has a hand model built-in, which is combined with the raw sensor data to track the positions and motion of the hand precisely. However, the effective range of this sensor is only 25 to 600 millimeters approximately which is not always suitable for the distant interactive applications between humans and robots.

III. Our Contributions

In this paper, we present a framework of pHRI using Kinect v2 with the deep learning APIs caffe, tensorflow and Keras. Moreover, we demonstrate the performance of our framework in a pHRI setting for a *Tool Handover Task* where inference from vision is combined with that from the torque sensors of a robotic arm. Our contributions in this perspective are summarized as follows:

1) Integration of the recent and accurate skeleton tracker *Openpose* [15], [16] with the Microsoft Kinect v2 (time-of-flight) sensor in Ubuntu 16.04 to get a robust 3D skeleton in real-time.
2) Development of a CNN for on-line hand gestures recognition at distances up to 4 meters from the robot. The CNN is trained on four gestures namely *Handover, Stop, Resume* and *None* gesture with post-processed hand images with extensive data augmentation to avoid any background removal or image processing in the recognition step.
3) Design of a pHRI framework combining vision and force measurements to achieve two-way object handover between operator and robot.

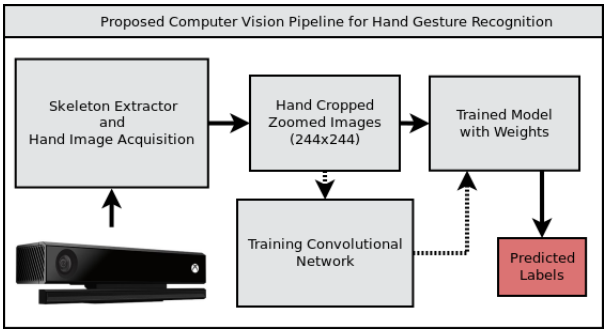

Fig. 2. Our proposed computer vision pipeline for hand gesture recognition.

IV. Methodology

The proposed pHRI pipeline is divided in three main modules namely: *Skeleton extraction and hand image acquisition*, *CNN for pHRI hand gestures recognition* and *Robot control for pHRI*. The proposed computer vision pipeline is illustrated in Fig. 2. The pHRI details on the modules are described in the following sections.

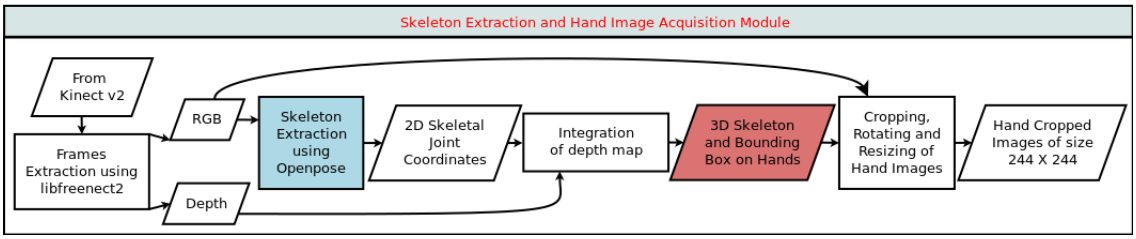

Fig. 3. Integration pipeline of Microsoft Kinect v2 with Openpose to extract human skeletal joint coordinates and hand images.

A. Skeleton Extraction and Hand Images Acquisition

To develop an efficient and human-friendly human-robot interaction setting, Microsoft Kinect v2 is opted for data acquisition. Opensource SDKs like OpenNi2 inherently provide skeleton extraction functions for Kinect v2, but they are not robust enough to be used in real-world pHRI settings. The latest developments in machine learning and availability of computational resources have allowed researchers to develop more robust techniques for this. Openpose is a recent development in this reference, which works on the basis of *Convolutional Pose Machines (CPMs)* [17]. It only requires RGB images to extract 2D skeletal joint coordinates of humans in the scene. Although Openpose can reliably extract the skeletal joints, the absence of the third coordinate i.e., the depth information, makes it inconvenient to be employed in pHRI scenarios. We integrate Kinect v2 with Openpose

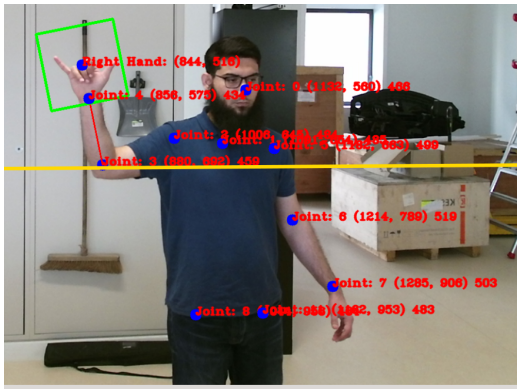

Fig. 4. 3D skeletal joints, hand center location and rotated bounding box (green). Hand gesture detection is invoked only when the forearm lies above the predefined yellow line.

to get the skeleton hands depth without compromising the computational cost. This information can be utilized to develop an efficient recognition system for both static and dynamic gestures. This also enables the robot to interact with the environment, particularly with the operator, using depth information.

Figure 3 illustrates our pipeline of integration of Kinect v2 with Openpose. To get an approximate location of the hand, we first fit a line (red in the figure) between the elbow and wrist joints and compute the angle that it makes with the vertical. This line is then extended to one third of the length of forearm to estimate the hand center location. Then we derive a bounding box (green in the figure), centered at the hand center and aligned with the forearm at all times. The size of the bounding box is determined by the mean depth value of 36 pixels (6×6 matrix) at the predicted hand center. This keeps the bounding box size close to that of the hand, irrespective of its distance from the Kinect v2 (obviously, within the sensor depth range). The extracted skeletal joints with their depth values, the line between the elbow and wrist joint, approximated hand center location and the rotated bounding box can be seen in Figure 4. The hand images are obtained by rotating and cropping the pixels that lie within the bounding box. This makes our system independent of hand orientation in the coronal plane of the human body. The cropped hand images are resized to 244x244 pixels images that are then fed to the CNN for gesture recognition.

Fig. 5. Trained hand gestures after data augmentation.

B. CNN for Hand Gestures Recognition

We develop a CNN to recognize the four hand gestures shown in Fig. 5. The architecture of our CNN, which is illustrated in Fig. 6 (top), is inspired mainly from LeNet with the addition of dropout layers, fully-connected layers and hyper-parameters tuning. The addition of dropout layers drastically improves the capability of the network to learn distinct features in the dataset. The dataset is generated for the four gestures performed by a single person, by recording the RGB and depth image stream from Kinect v2. The saved

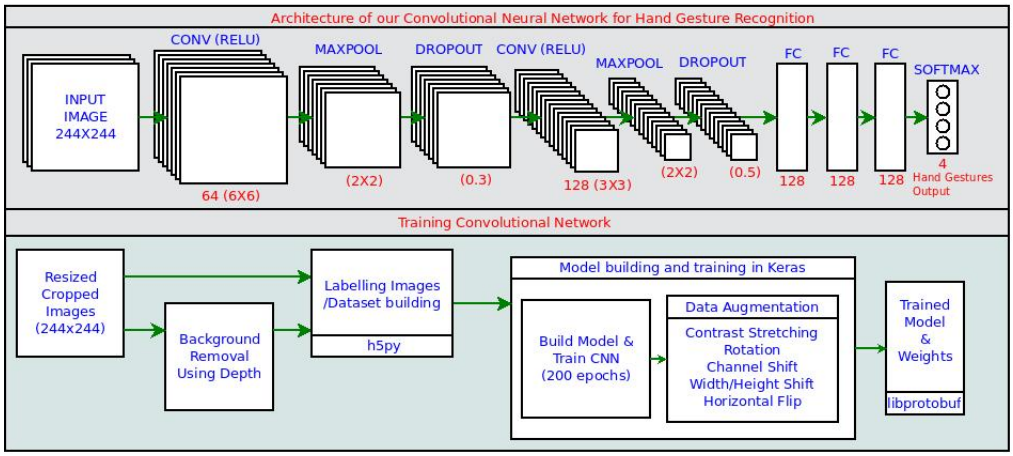

Fig. 6. Architecture of our Convolutional Neural Network (top), and block diagram of training and building CNN model (bottom)

images are then passed through the skeleton extraction and hand images acquisition module as explained in the previous section and hand images are stored and later manually copied into different folders. These images are then labeled and stored in a h5 file for later use in model training.

To make gesture recognition invariant to background, we also train the CNN with augmented backgrounds. Let us now detail our *background augmentation* approach. A binary mask for background subtraction is created using the depth information from Kinect v2. It is created such that the pixels that lie at a depth within ± 18 % (empirical value) of the mean depth value computed at the wrist joint (obtained through openpose) are forced to the value 1, while the rest are zeroed. This binary mask is broadcasted into three channels

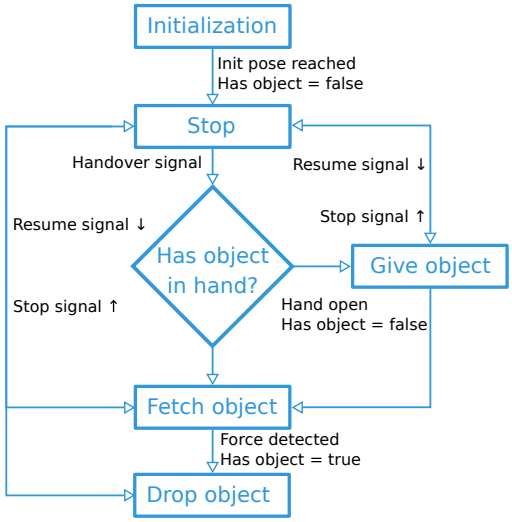

Fig. 7. Finite state machine of the robotic handover experiment.

and then multiplied by the cropped RGB hand image. This mean depth value is computed as in the case of estimating the bounding box size over the hand (Sect. IV-A). Then, each image in this pre-processing loop is assigned a different gray value as background. This is achieved by first creating an inverted binary mask by simply applying a "NOT" operation on the mask originally computed. This inverted mask is multiplied with gray values incremented for each image by 10, broadcasted into three channels and then added with the background-removed hand image. To improve the network invariance to background even further, different patterns can also be introduced in the background. This step is left for the future work and is not performed in the current system. Other data augmentation techniques that we applied to the database are: contrast stretching, channel shift and horizontal flip. The results of data augmentation are shown in Fig. 5. A block diagram of training our CNN is shown in Fig. 6 (bottom). The Keras python API is used to build the network and for data augmentation and network training. The network is trained overnight on a set of 1800 RGB images of size 244x244 pixels on an Intel Core i7-6800K CPU at 3.40GHz, 12 cores with no GPU. Validation accuracy, with 600 test images, is 98.8 %.

The trained model is converted into a protobuf file, to be later used with the tensorflow C++ library in the online recognition phase. We evaluated our model with 300 more test images extracted from a video recorded in different light conditions, and achieved 95.7 % accuracy on these data. We plan to extend this system by collecting more data from multiple users and to test the accuracy of our network on persons not included in the data. We have performed the initial tests in this reference, however we will quantify and publish these results in future.

C. Robot Control for pHRI

The BAZAR robot used for the experiments is composed of two Kuka LWR 4+ arms with two Shadow Dexterous Hands attached at the end-effectors. The arms are attached to a Neobotix MP700 omnidirectional mobile platform. In our scenario, the mobile base is kept fixed and only the right hand-arm system is used. The control of the arm is done using the FRI library and the control of the hand is based on a ROS interface. The external force applied to the arm's end-effector is estimated by FRI based on joint torque sensing

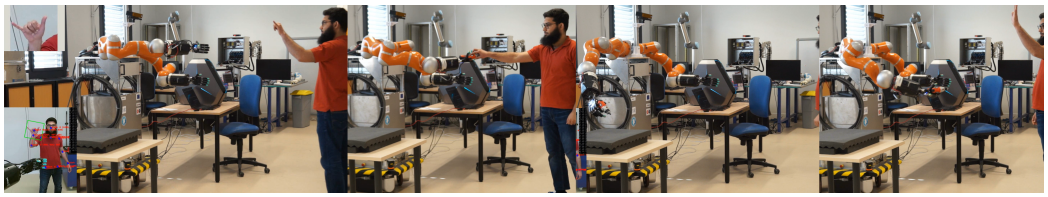

Fig. 8. Screen-shots of the video of experiment that we performed to analyze the robustness of the proposed pHRI framework. Starting from left, the human operator is giving a *Handover* command to the robot, then the operator hands over the tool to the robot, the robot then moves toward the table and drop the object, the operator is giving a *Stop* command to the robot in the last frame. The video can be accessed through the link in the footnote.

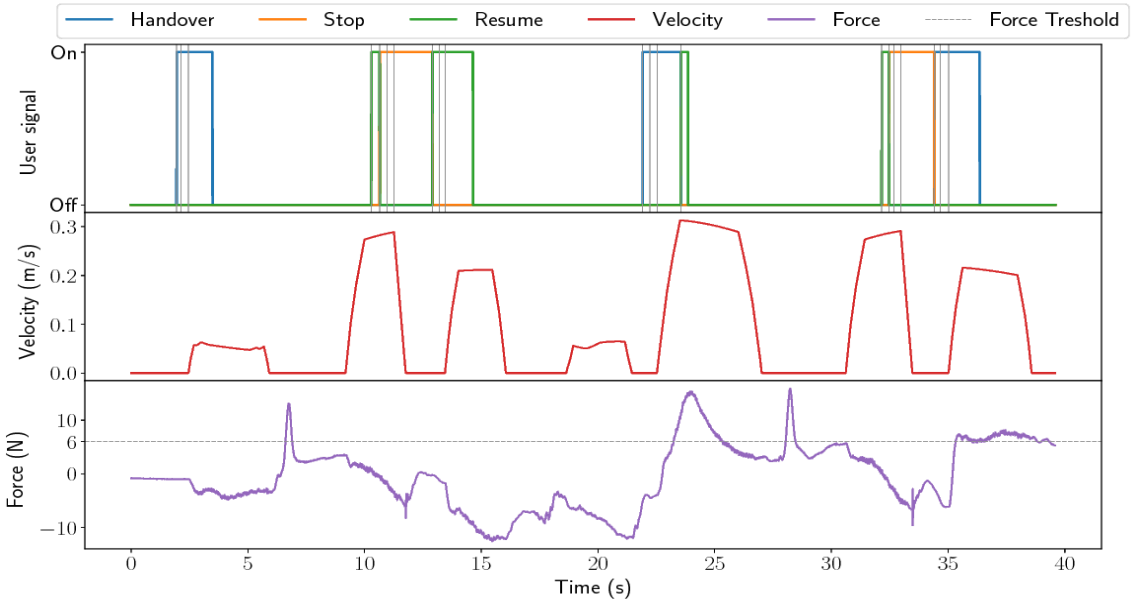

Fig. 9. Illustration of detected user inputs (top), l2-norm of end-effector velocity (center) and l2-norm of force applied on the end-effector (bottom).

GESTURES SEQUENCE →	HANDOVER	STOP	RESUME	HANDOVER	STOP	HANDOVER
	DETECTION TIME INSTANCES (in sec)					
	1.960	10.660	10.300	21.926	32.475	34.420
	2.135	10.970	12.915	22.235	32.680	34.665
	2.470	11.275	13.220	22.540	32.985	35.035
			13.466			
			23.565			
			32.165			
GDRT (sec)	0.510	0.615	0.551	0.614	0.510	0.615

Fig. 10. Time instances of the detected user inputs (as shown in Fig. 9(top) as rising edges of colored plots and gray lines. The red circled time instances are false positives, which are detections that happen less than three times in succession hence not taken into account by the robot controller. Gesture detection response time (GDRT) comes from subtracting third time instance from the first in a successful detection (in green boxes).

and on knowledge of the robot's dynamic model. The control rate is set to 5ms.

V. PHRI Experiment and Results

For safe pHRI, the robot must perceive the intention of the operator. Here, 3D human body joint coordinates and hand gesture recognition are the cues used for robot operation. We realize a tool (here, a portable screw-driver) handover experiment, guided by the finite state machine presented in Fig. 7. The robot waits for the user commands in the form of hand gestures, to take and then place the tool to a predefined location in its workspace. In Fig. 8 we show some screen-shots of the experiment, and a complete video is attached to this paper, and can be accessed through this link [1]. The commands are fulfilled by the robot when three successive identical instances of the corresponding gesture are detected, and only if the forearm is above the horizontal line passing through the elbow joint (see Fig. 4). This aids in ignoring all gesture detections when the operator does not intend to interact with the robot and has relaxed his/her arm. This can also be seen in Fig. 9(top), where gray vertical lines after

[1] http://bit.do/d8ukg

each user-input detection represents successive detections of the same gesture.

Interaction is started by detection of the *Handover* command, which triggers the motion of the arm-hand toward the operator and the opening of the fingers, in a predefined configuration, suitable to carry the object. As the robotic arm moves toward the operator, a *Stop* command can stop this movement and the robot keeps the halt position until a *Resume* command is received. Moreover, if the robot has received the object and is moving toward the table to place the object on top of it, a *Stop*, followed by a *Handover* command will make the robot return the object to the operator. The velocity curve in Fig. 9 is aligned with the user-signal detection plot, so the motion of the end-effector corresponds to the detected input. A threshold of 6 N in the X (downwards) component of the force applied on the end-effector is used to trigger tool grasping. This can be seen after the detections of *Handover* commands in Fig. 9. The robot continues to execute the previous command, even when a *None* gesture is detected.

This experiment is performed indoor and all gesture permutations are tested. The operator moves closer and farther from the robot and is allowed to move his hand in the coronal plane depending on his comfort. The robot is able to detect and obey the intended commands within approximately 570 milliseconds from a single operator in the scene. We call it gesture detection response time (GDRT) which is an average time combining three successive detections of hand gestures. The detection time instances of the gestures recognized in this interaction experiment are presented in Fig. 10. Our overall pHRI framework is able to extract 3D skeleton joint coordinates along with the detection of hand gestures with an approximate frame-rate of 5.2 fps (approximately 192 millisecond for a single execution of our framework loop). The Openpose skeleton extractor is the main bottle-neck in the pipeline, since it already requires a GPU (GeForce GTX 1080 in our case) to execute pose extraction using caffe. The forward-pass of input hand images through our tensorflow model requires no GPU at this moment. Nevertheless, our CNN model can be trained and run using a GPU in a multiple GPU hardware for faster recognition rates. Since multiple GPUs can also be used in the provided Openpose wrapper with supported cameras, we are confident that in the near future we will increase hand gesture detection rates and ensure human-like speed.

VI. Conclusion

For safe and intuitive pHRI, availability of sufficient human-body descriptors is imperative for a robot to successfully understand and obey the intended gestures. A robust pHRI framework has been developed and presented in this paper. It includes extraction of a 3D human skeleton through integration of a Kinect v2 with a state-of-the-art 2D skeleton extraction library and a fast hand gesture recognition system. A database collection, including user study, has already been developed and is in execution while we write this paper. This will allow us to incorporate more gestures from many different people in our framework. The descriptors from 3D human-skeleton will be extracted and used to recognize dynamic gestures in a human-robot interaction scenario for a more natural and productive cooperative activity. Multiple objects detection, their localization in the scene and handling of objects with different shapes will be added in future work. The use of recurrent neural networks for the detection of dynamic gestures/human-intention will be explored and will also be added to our pipeline.

References

[1] A. Mehrabian. *Nonverbal Communication*. Aldine Publishing Company, 1972.

[2] G. Canal, S. Escalera, and C. Angulo. A real-time human-robot interaction system based on gestures for assistive scenarios. *Computer Vision and Image Understanding*, 149:65–77, 2016.

[3] Hongyi Liu and Lihui Wang. Gesture recognition for human-robot collaboration: A review. *International Journal of Industrial Ergonomics*, 2017.

[4] S. Mitra and T. Acharya. Gesture recognition: A survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 37(3):311–324, May 2007.

[5] Grazia Cicirelli, Carmela Attolico, Cataldo Guaragnella, and Tiziana D'Orazio. A kinect-based gesture recognition approach for a natural human robot interface. *International Journal of Advanced Robotic Systems*, 12(3):22, 2015.

[6] Jagdish Lal Raheja, Mona Chandra, and Ankit Chaudhary. 3d gesture based real-time object selection and recognition. *Pattern Recognition Letters*, 2017.

[7] Y. LeCun, Fu Jie Huang, and L. Bottou. Learning methods for generic object recognition with invariance to pose and lighting. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 2, pages II–97–104 Vol.2, June 2004.

[8] Pei Xu. A real-time hand gesture recognition and human-computer interaction system. *CoRR*, abs/1704.07296, 2017.

[9] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, Nov 1998.

[10] K. Ehlers and K. Brama. A human-robot interaction interface for mobile and stationary robots based on real-time 3d human body and hand-finger pose estimation. In *2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 1–6, Sept 2016.

[11] Y. Yang, H. Yan, M. Dehghan, and M. H. Ang. Real-time human-robot interaction in complex environment using kinect v2 image recognition. In *2015 IEEE 7th International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM)*, pages 112–117, July 2015.

[12] R. C. Luo and Y. C. Wu. Hand gesture recognition for human-robot interaction for service robot. In *2012 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 318–323, Sept 2012.

[13] M. Van den Bergh, D. Carton, R. De Nijs, N. Mitsou, C. Landsiedel, K. Kuehnlenz, D. Wollherr, L. Van Gool, and M. Buss. Real-time 3d hand gesture interaction with a robot for understanding directions from humans. In *2011 RO-MAN*, pages 357–362, July 2011.

[14] M. Munaro, F. Basso, and E. Menegatti. Tracking people within groups with rgb-d data. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2101–2107, Oct 2012.

[15] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017.

[16] Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *CVPR*, 2017.

[17] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. *CoRR*, abs/1602.00134, 2016.