Enhancing Pepper Robot's Human-Robot Interaction Capabilities through Advanced Hardware Integration and Human Pose Estimation

Paolo Magri, Javad Amirian, and Mohamed Chetouani

Sorbonne Université, CNRS, Institut des Systèmes Intelligents et de Robotique (ISIR), Paris, France
[magri, amirian, chetouani]@isir.upmc.fr

Abstract. In this paper we propose hardware and software enhancements for the Pepper robot to improve its human-robot interaction capabilities. This includes the integration of an NVIDIA Jetson GPU to enhance computational capabilities and execute real-time algorithms, and a RealSense D435i camera to capture depth images, as well as the computer vision algorithms to detect humans around the robot and estimate their body orientation and gaze direction. We have also collected a MoCap dataset of human activities in a controlled environment, together with the corresponding RGB-D data to validate the proposed algorithms. Additionally, we provide the ROS stack and communication schematics between different nodes to facilitate a legible and more efficient navigation system.

Keywords: Field Of View Estimation · Depth · GPU · ROS Stack

1 Introduction

The integration of social robots into daily environments has progressed significantly, necessitating advancements in human-robot interaction (HRI). This paper builds on the existing body of work [1] [2] by focusing on the Pepper robot , a widely used social robot, and explores both hardware and software enhancements to improve its interaction capabilities. The hardware upgrades are explained in detail in the second paragraph. These enhancements are designed to support advanced object detection and pose estimation algorithms, and on top of that to calculate **MC:** *human* gaze and body orientation, thus enabling a better HRI in real-world scenarios. Our core contributions include:

1. Detailed instructions for implementing hardware upgrades on the Pepper robot, including the installation of an NVIDIA Orin Jetson Nano GPU and a RealSense D435i RGB-D camera.
2. Integration in ROS of the software for human detection and **MC:** *Field Of View* estimation with the robot's firmware, obtaining a flexible system.
3. The development of a comprehensive dataset for testing and improving the robot's interaction algorithms.

2 Hardware Upgrades to the Pepper Robot

Trying to improve the HRI of the Pepper robot, we recognized the necessity to upgrade its hardware components to meet the increasing demands of advanced deep learning models. The original hardware of the Pepper robot does not possess the required power to support our proposed software [2].

Its Intel Atom processor lacks the necessary processing capacity to execute the perception and planning algorithms in real-time. Additionally, the robot's camera, with its low resolution and frame rate, is inadequate for environmental capture. To overcome these limitations, we implemented several hardware upgrades to the robot. The NVIDIA Orin Jetson Nano Development Kit connects via Ethernet cable, a decision driven by its robust specifications tailored for AI applications. This upgrade significantly enhances the computational capabilities of the Pepper robot [2], details of which are as follows: As a secondary enhancement, we have integrated the Intel RealSense D435i camera into the Pepper robot's hardware configuration. Given budget constraints and the requirements of our applications, the RealSense presents an optimal balance between cost and performance. This camera is pivotal for enabling 3D depth estimation providing detailed depth information and it is particularly beneficial for the robot's interaction with its surroundings, enhancing its ability to detect and navigate around obstacles and individuals.

Table 1. Comparison of Nvidia Orin Jetson Nano [3],Intel RealSense D435i[5], new LiPo and Pepper Robot Hardware[6]

Component	**Original Hardware**	**New Hardware**
GPU	None	NVIDIA Ampere architecture GPU, 1024 CUDA cores, 32 tensor cores, Max Frequency 625MHz
CPU	Intel Atom E3845, Quad-core 1.91 GHz	6-core Arm Cortex-A78AE v8.2 64-bit processor, 1.5MB L2+4MB L3 cache
Camera	30 FPS	Up to 90 FPS RGB + 30 FPS depth
Battery	2950 mAh	Previous Battery + 2200 mAh

Finally, a 2200 mAh 11.1V LiPo battery has been added, which can power the new hardware for at least 1.10 hours [1].

Fig. 1. Pepper Robot with New Hardware Installed

CAD Design for Hardware Integration

To accommodate the NVIDIA GPU and the enhanced battery system, we have designed specialized 3D printed parts; the structural modifications have been optimized to withstand the operational demands placed on the robot, ensuring that it can perform its duties without any hardware-related disruptions.

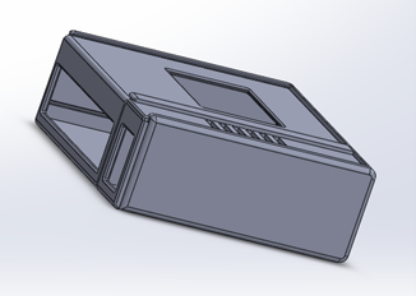

Fig. 2. Custom-designed box has been created to house both the GPU and the battery. This enclosure is equipped with air vents to ensure adequate cooling, essential for maintaining optimal performance and longevity of the hardware. The design considerations also include ease of access for maintenance.

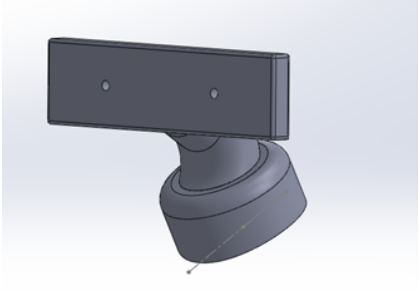

Fig. 3. RealSense D435i Camera Mount has been developed sturdy and compact. This mount is designed to be thick and short to minimize vibrations. The stability provided is crucial for maintaining the precision of the depth measurements during dynamic interactions and movements.

3 Human Detection and FOV Estimation

The use of advanced deep learning techniques, such as YOLOv8 and pose analysis, has significantly enhanced the ability of robots to interact with humans, improving the effectiveness of these technologies in real-world scenarios.

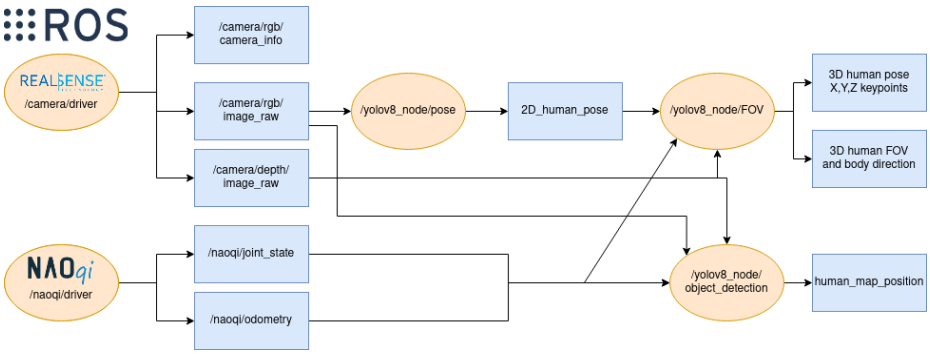

Fig. 4. ROS stack for communication between NAOqi, Realsense, and YOLO

Application of YOLOv8 and YOLOv8 Pose in Human Detection
YOLOv8 segments images into regions, identifying and classifying present objects with a single analysis. The YOLOv8 Pose variant extends these capabilities to pose estimation, detecting human keypoints such as joints and body segments. This precision is crucial for accurately interpreting human postures and actions.

Improvements in Keypoint Detection
Using convolutional neural networks (CNN), YOLOv8 Pose identifies significant human body keypoints in real time, such as eyes, shoulders, and knees. These keypoints are analyzed to compute the spatial configuration of the body, allowing robotic systems to interpret human postures and movements with high precision. To mitigate accuracy issues in depth measurement, essential for precise three-dimensional localization, several approaches are adopted:

- Distinction between background and foreground: for each human's keypoint, a circle with a radius of 5 is calculated, and the minimum depth value is selected to reduce noise and measurement discrepancies.

- Correction Algorithms: The Kalman Filter improves the consistency and accuracy of FOV estimation in critical situations (e.g., when a person is in profile relative to the camera).

Determining Body Orientation and Gaze Direction
The orientation of the torso,for each frame, is defined by calculating the average vector between the keypoints of the shoulders and hips on a horizontal plane. Similarly, the direction of gaze is based on the keypoints of the eyes and neck.
1. The normal vector $\mathbf{N}$ to the plane formed by vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ is given by the cross product:

$$\mathbf{N}_{\text{torso}} = (\mathbf{P}_{\text{shoulder.L}} - \mathbf{P}_{\text{pelvis}}) \times (\mathbf{P}_{\text{shoulder.R}} - \mathbf{P}_{\text{pelvis}}) \tag{1}$$

$$\mathbf{N}_{\text{gaze}} = (\mathbf{P}_{\text{eye.L}} - \mathbf{P}_{\text{neck}}) \times (\mathbf{P}_{\text{eye.R}} - \mathbf{P}_{\text{neck}}) \tag{2}$$

2. The projection of the normal vector onto the XY-plane and the normalized direction vectors for the torso and the gaze are:

$$\mathbf{d}_{\text{torso}} = \frac{\mathbf{N}_{\text{torso},xy}}{\|\mathbf{N}_{\text{torso},xy}\|}, \quad \mathbf{d}_{\text{gaze}} = \frac{\mathbf{N}_{\text{gaze},xy}}{\|\mathbf{N}_{\text{gaze},xy}\|} \tag{3}$$

3. The quaternion representing the rotation from the forward direction $[1, 0, 0]$ to the body direction vector and the view direction vector is:

$$\mathbf{q}_{\text{torso}} = \left[\cos\left(\frac{\theta_{\text{torso}}}{2}\right), \mathbf{u}_{\text{torso},x}\sin\left(\frac{\theta_{\text{torso}}}{2}\right), \mathbf{u}_{\text{torso},y}\sin\left(\frac{\theta_{\text{torso}}}{2}\right), \mathbf{u}_{\text{torso},z}\sin\left(\frac{\theta_{\text{torso}}}{2}\right)\right] \tag{4}$$

$$\mathbf{q}_{\text{gaze}} = \left[\cos\left(\frac{\theta_{\text{gaze}}}{2}\right), \mathbf{u}_{\text{gaze},x}\sin\left(\frac{\theta_{\text{gaze}}}{2}\right), \mathbf{u}_{\text{gaze},y}\sin\left(\frac{\theta_{\text{gaze}}}{2}\right), \mathbf{u}_{\text{gaze},z}\sin\left(\frac{\theta_{\text{gaze}}}{2}\right)\right] \tag{5}$$

We have adopted a 120-degree horizontal human FOV in accordance with [4], to accurately map what the person can see.

4 MoCap Dataset Acquisition and Algorithm Validation

The objective of the dataset is to acquire keypoints of a person, using a motion capture system with 8 cameras in a closed environment with standard lighting, simulating various types of human movements, capturing different distances, orientations, heights, and poses. The following protocol was defined for the trials, each lasting approximately 2 minutes with a single person in the room. It is assumed that the algorithm will function in multi-person scenarios, without considering the loss of information due to occlusion of the camera's field of view by objects and people. The various scenes were also recorded simultaneously with a Realsense camera, positioned statically at a height of 1.25 meters to obtain the same view as when Pepper is in an upright position.

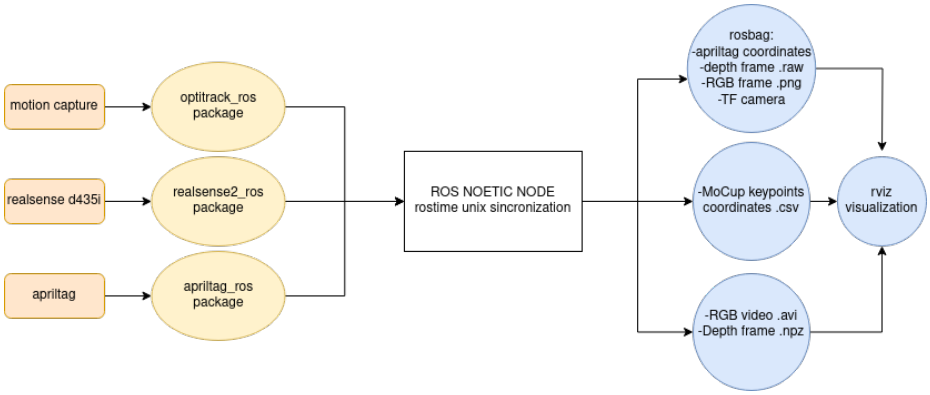

Fig. 5. Dataset acquisition schema for testing FOV Estimation using MoCap as ground truth. Apriltag and the camera's TF are used for spatial calibration, while ROS time is used for temporal synchronization.

The analyzed scenes include:

- Standard walk back and forth
- Walk back and forth with arms crossed
- Walk back and forth with sudden movements (e.g., dodging an obstacle)
- Walk back and forth in a zigzag pattern with pronounced head movements

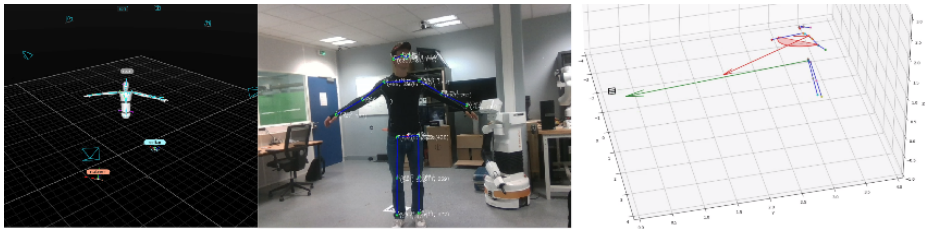

Fig. 6. A.Skeleton generated from MoCap B.RGB image from RealSense with YOLOv8 pose C.skeleton with FOV (red) and body direction (green) generated by algorithm

5 Conclusion and Implementation in Real Scenario

To conclude, we analyze how communication between different algorithms via ROS should be structured to enable the robot to perform a task involving navigation, such as transporting an object from point A to point B or moving to a specific location. Below, all the nodes that enhance navigation readability are shown, thereby improving the robot's ability to understand and be better understood by the humans present in the environment.

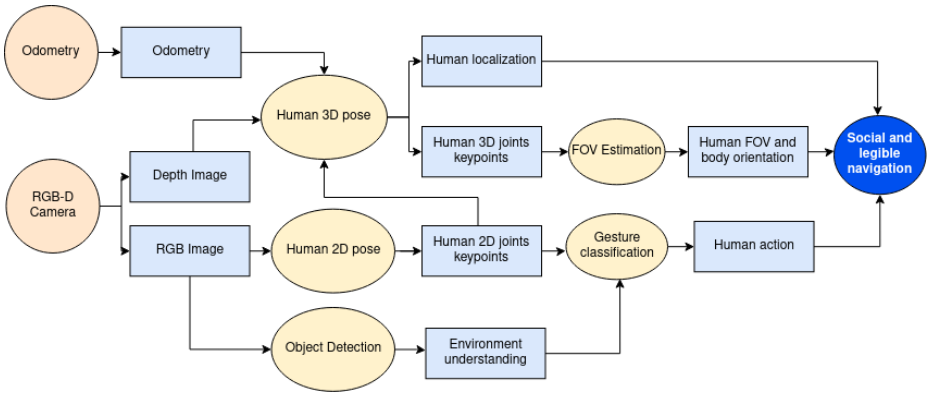

Fig. 7. Full schema to achieve readable navigation thanks to the understanding of humans surrounding the robot.

6 Acknowledgments

This work has received funding from the European Union's Horizon 2020 research and innovation programmes under grant agreement No 952026 and Horizon Europe Framework Programme under grant agreement No 101070596.

References

1. J. Amirian, M. Abrini, and M. Chetouani, "Legibot: Generating Legible Motions for Service Robots Using Cost-Based Local Planners," arXiv e-prints, p. arXiv–2404, 2024.
2. M. Caniot, V. Bonnet, M. Busy, T. Labaye, M. Besombes, S. Courtois, and E. Lagrue, "Adapted pepper," arXiv preprint arXiv:2009.03648, 2020.
3. NVIDIA, *Jetson Orin*, https://www.nvidia.com/, accessed: 2024-06-01.
4. A. V. Taylor, E. Mamantov, and H. Admoni, "Observer-aware legibility for social navigation," in 2022 31st IEEE International Conference on Robot and Human Interactive Communication (RO-MAN). IEEE, 2022, pp. 1115–1122.
5. Intel, *Intel RealSense Depth Camera D435i*, https://www.intelrealsense.com/depth-camera-d435i/, accessed: 2024-07-10.
6. SoftBank Robotics, *Pepper Robot Hardware Specifications*, https://www.softbankrobotics.com/emea/en/pepper, accessed: 2024-07-10.