RoboBreizh 2021 Team Description Paper

A. DIZET [1] C. LE BONO [1] A. LEGELEUX [1] M. NEAU [1]
N. WONDIMU [1] S. RASENDRASOA [2] Y. OMAR [4] M. BOUABDELLI[2]
A. PAUCHET [2] D. DUHAUT [1] C. BUCHE [3]

March 5, 2021

Abstract. Our team, RoboBreizh, was founded in 2018. In 2020, we have won the Best Performance award at RoboCup@Home EDU. Currently, we have 11 members from four different laboratories based in France and Australia. This paper aims to introduce the activities that are performed by our team and the technologies that we use. Main contributions include efficient detection, new NLP pipeline and gestures learning by demonstration. Our team is able to work using real robot, qiBullet or Gazebo simulator. RoboBreizh develops his own Pepper Gazebo full environment.

1 Introduction

RoboBreizh is initialy a RoboCup French team of the Brest National Engineering School (ENIB). The team was founded in 2018. Since then, RoboBreizh has won the Best Performance award at RoboCup@Home EDU competitions in 2020. RoboBreizh became a joint French team between the ENIB and the National Institute of Applied Sciences (INSA) Rouen Normandy.

The 2021 team consists of the following persons :
Students: Maëlic NEAU, Antoine DIZET, Amélie LEGELEUX, Cédric LE BONO, Natnael WONDIMU, Sandratra RASENDRASOA
Post-doctorate & Engineer: Yasser OMAR, Maël BOUABDELLI
Leaders: Cedric BUCHE, Alexandre PAUCHET, Dominique DUHAUT
Website: https://www.enib.fr/∼robobreizh

This paper is divided as follow. Section 2 presents RoboBreizh's main research innovations. Next, section 3 describes the architecture and the platforms proposed. Section 4 focuses on perception, navigation, movement and human-robot interaction (NLP and gestures). Finally, section 5 concludes this article.

[1]Lab-STICC, France
[2]LITIS, France
[3]IRL CROSSING, CNRS, Australia
[4]CCIT, AATMT Cairo branch, Egypt

2 Team research focus

The team offers original and efficient solutions in various contexts. Notably, our robot perception is handled using state-of-art algorithms to detect people and objects. Pepper moves its arms to support natural interaction and picks up objects using a Learning by Demonstrations model. The NLP part offers an efficient pipeline combining various complex modules. The team works both on real environment and virtual environment (qiBullet/Gazebo).

3 Architecture and environment

3.1 System Architecture

Our architecture is built upon 5 ROS modules: Manager, Perception, Navigation, Movement and Interaction. The manager is a high level structure. Modules output are stored as object instances and available as input information, later. The manager executes tasks in the required order, scheduling Pepper behavior. It also handles task priority. This architecture was designed to easily include new robot behaviors, without editing any external modules. In order to bypass execution lags, orders are cancelled if they take too long to execute.

3.2 Platform

Our system has been tested using real and simulated environments. Concerning simulation two different simulators are used. First, qiBullet [1] is a new simulation environment provided by SoftBank Robotics for Pepper and Nao robot. The advantage of qiBullet is that it can emulate the Pepper system NAOqi. Moreover, this simulator provides a ROS wrapper that makes it possible to run our code. Then, we also use the simulator of ROS, Gazebo [2] (version 7) to emulate the Pepper robot (meshes and sensors) but not the NAOqi system.

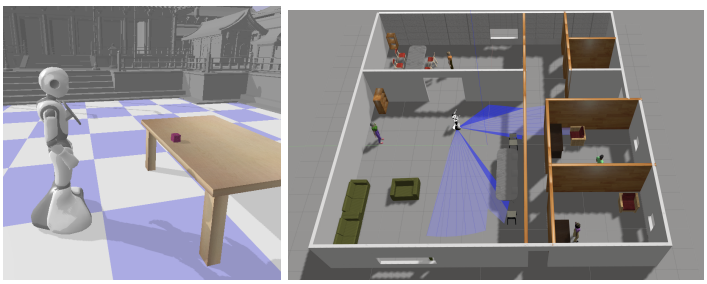

Fig. 1. 3D simulator: (left) QiBullet [1] and (right) Gazebo [2]

4 Approaches

4.1 Perception

YOLO [3] has become a standard in computer vision, especially for object and person detection. Other solutions are available, such as Mask-RCNN [4] that provides a mask generation of objects detected in addition to the bounding boxes. Beyond object and people detection, performing pose estimation was deemed necessary to detect people's movement (e.g. waving hands). An efficient tool in this domain is OpenPose [5], a real time multi-person system which can detect up to 135 different kinds of body keypoints. Mask R-CNN was chosen over other state-of-the-art object detection algorithms due to its ability to detect objects with pixel-level precision. Compared to other solutions, Mask R-CNN minimizes the noise added by the background and reduces the risk of inaccurate localisation of object. This level of precision is required when measuring the distance between Pepper and an object. Our module is also capable of understanding the current state of objects and person. For instance, by using the positions of chairs and persons in an image and how they overlap, it is possible to determine whether the chairs are available. Additionally, OpenPose is exploited to extract the positions of all the hands in an image and thus whether someone is waving. Also, gender and age estimation is performed using models proposed by [6].

4.2 Navigation

Our approach for navigation is based on a Pepper specific implementation of the ROS Navigation Stack [7]. First, mapping is throughout Gmapping ROS package which provides a tool to generate 2D occupancy map from laser sensors data using SLAM. Unfortunately, data provided by Pepper laser sensors is not sufficiently accurate to build a detailed map. Consequently, in addition to those inputs, we decided to feed the mapping node with data from the Pepper RGB-D camera. Once a valid map is obtained, it can be used for navigation. Thus, a real-time localization is performed using amcl ROS node [8]. The next step computes a path through a given goal and achieve it. This is performed by move_base Node [9] that defines global and local planner for the robot to follow.

4.3 Movement

Robots move their bodies to interact with theirs environments and with humans. Learning by demonstrations is one of the easiest way to teach a movement to a robot. Kinesthetic demonstrations (a human moves the robot arms) are exploited. With multiple demonstrations, the robot can generalize the movement. The learning is done at the trajectory level. We use Gaussian Mixture Model (GMM) and Gaussian Mixture Regression (GMR) to learn a movement with multiple demonstrations [10]. The initialization of the means is done with K-means algorithm and the selection of the number of Gaussians with the BIC score. The movement module is developed under some constraints. Pepper can

only take light objects because of its hands. It also has limited movements due to the robot reachable workspace. This module is developed to simplify movement learning. The learning movement module is composed of two parts: the learning phase and the movement phase. In the learning phase, the user can make multiple demonstrations to the robot for a single movement. Then the learning algorithm uses previous demonstrations. Figure 2 shows the generalization of the movement "Point a seat" of the joint 9 (corresponding to the right shoulder roll) with two demonstrations. The movement phase generates the movement with the result of the GMR. The movement module can replicate the learned movement in real time as the learning phase was previously performed. The movement cannot be adapted to any environment because the model used is a simple GMM/GMR. The modified GMM/GMR [11] can overcome this constraint and add an obstacle avoidance skill.

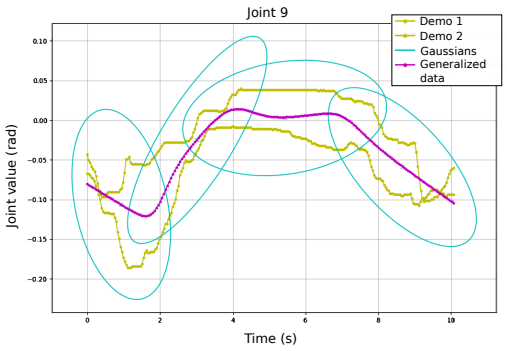

Fig. 2. Learned movement "Point a seat" of the joint 9 with the GMM/GMR. The two demonstrations are in green, the learned movement is in pink. The four Gaussians are displayed in blue.

4.4 Human-Robot Interaction (HRI)

Speech Recognition and NLP

RoboBreizh proposes a new pipeline designed upon Speech Recognition and Natural Language Processing (NLP). This pipeline is connected to Naoqi via a system detecting the user's input voice by analysing the evolution of ambient sound level. In our proposal, the Speech Recognition is handled by the Google API (Google Cloud Speech-To-Text) and NLP by using learning approaches such as Mbot [12] combined with rule-based APIs (e.g. Spacy [13]). In addition to those solutions, a Dialog Act classifier ([14]) is exploited. This classifier enables to adapt the system response to the type of dialog. For example, if the utterance is considered as an Action-Command the system runs Mbot as an intent analysis, otherwise the classic rule-based model is used. Using this classifier

as a pre-processing unit saves time and prevent intent classifier mistakes. We also implement a Sentiment Analysis Module [15] to classify user's utterance between "negative", "neutral" and "positive" sentiment. This enables to adapt Pepper's response using a rule-based system. An additional component enables to better understand the user intent through the use of commonsense knowledge base and deep-learning based pipeline [16]. Information inside a knowledge base can be represented in a tuple format e1, r, e2 where e1 and e2 are two entities in a relation r. An example of tuple related to the sentence "Going outside" would be: (E1 = "Go outside", R = Causes, E2 = "feel better"). In this example, the objective would be to generate E2 given E1 and R. First, we have selected specific relationships from the ConceptNet database. Then, through using the COMET model, we are able to generate common sense facts given the user's command.

Speaker recognition

We adapted classic techniques that works directly on the raw signal data without the need of handcrafted features [17]. The proposed model exploits SincNet, which requires as learning parameters only lower and higher cut frequencies, and therefore reduces the number of parameters learned per each filter and makes this number of parameters independent of the range of each filter. In addition, we combined both the SincNet and a Siamese Network with an algorithm to train Siamese neural networks in speaker identification. The algorithm is funded on the selection of the best anchor for each class. In addition, preparing negative pairs is done based on pairs that are nearest to the anchor class in features space. This selection enhances the performance of the Siamese network since it ensures to learn the confusing cases.

5 Conclusion

This paper describes the RoboBreizh team approach. A ROS architecture was developed to handle the competition tasks using a Pepper robot. Notably, our robot can move to a destination, point to an empty seat, take a bag in hand, compute a person's position, talk with someone and detect a waving hand. Our proposed architecture is also flexible and can be easily implemented on other robots. In the future, the team will develop additional features to increase Pepper usability in user-friendly environments.

Acknowledgments

This article benefited from the support of the project Prog4Yu ANR-18-CE10-0008 of the French National Research Agency (ANR), the French National Centre for Scientific Research (CNRS) and the INCA project (Natural Interactions with Artificial Companions) of the Normandy region. We also thank the City of Brest (BMO), CERVVAL company, Australian-French Association for Research and Innovation (AFRAN), Brittany Region and the National Engineering School of Brest (ENIB) for supporting this project.

References

1. Maxime Busy and Maxime Caniot. qibullet, a bullet-based simulator for the pepper and nao robots. *arXiv preprint arXiv:1909.00779*, 2019.
2. N. Koenig and A. Howard. Design and use paradigms for gazebo, an open-source multi-robot simulator. In *International Conference on Intelligent Robots and Systems (IROS)*, volume 3, pages 2149–2154 vol.3, 2004.
3. Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
4. Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. *CoRR*, abs/1703.06870, 2017.
5. Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *CoRR*, abs/1812.08008, 2018.
6. Gil Levi and Tal Hassner. Age and gender classification using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 34–42, 2015.
7. Vittorio Perera, Tiago Pereira, Jonathan Connell, and Manuela M. Veloso. Setting up pepper for autonomous navigation and personalized interaction with users. *CoRR*, abs/1704.04797, 2017.
8. Brian P. Gerkey. amcl - ros wiki. http://wiki.ros.org/amcl, 2015.
9. E. Marder Eppstein. move_base - ros wiki. http://wiki.ros.org/move_base, 2016.
10. S. Calinon. *Robot Programming by Demonstration: A Probabilistic Approach*. EPFL/CRC Press, 2009.
11. Maria Kyrarini, Muhammad Abdul Haseeb, Danijela Ristić-Durrant, and Axel Gräser. Robot learning of industrial assembly task via human demonstrations. *Autonomous Robots*, 43(1):239–257, 2019.
12. Erick Romero Kramer, Argentina Ortega Sáinz, Alex Mitrevski, and Paul G Plöger. Tell your robot what to do: evaluation of natural language models for robot command processing. In *Robot World Cup*, pages 255–267. Springer, 2019.
13. Matthew Honnibal and Ines Montani. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear, 2017.
14. Sujith Ravi and Zornitsa Kozareva. Self-governing neural networks for on-device short text classification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 887–893, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.
15. C.J. Hutto and Eric Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. 01 2015.
16. Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Çelikyilmaz, and Yejin Choi. Comet: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.
17. Mirco Ravanelli and Yoshua Bengio. Speaker recognition from raw waveform with sincnet. In *Spoken Language Technology Workshop*, pages 1021–1028. IEEE, 2018.