Walking Machine @Home 2019 Team Description Paper

Jeffrey Cousineau, Huynh-Anh Le, et al.

École de Technologie Supérieure
1100 rue Notre-Dame Ouest, Montreal, QC, Canada H3C 1K3
http://walkingmachine.ca, walking@ens.etsmtl.ca,
https://github.com/WalkingMachine

Abstract. This paper gives details about the RoboCup@Home league team Walking Machine, from ETS University in Montreal, Canada for the next competition in Sydney, in July 2019. The robot from Walking Machine, named S.A.R.A. for "Systeme d'Assistance Robotique Autonome" (in English, Automated Robotic Assistance System), is a robot entirely built by the scientific club from ETS, mainly composed of undergraduates students. The robot is used for social interaction with humans, navigation and object manipulation. This document shows the electrical, mechanical and software novelties and functionalities of S.A.R.A.

1 Introduction

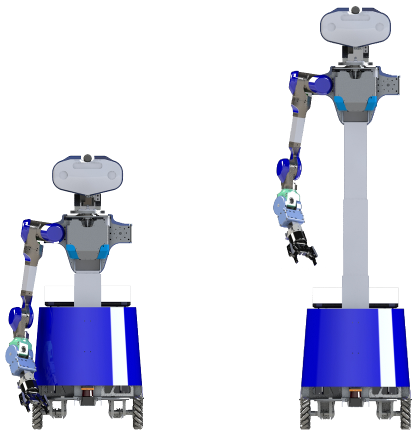
Walking Machine's team is a young team from Montreal, Quebec, in Canada, composed of engineering students in the field of mechanical, electrical and software engineering. We have been working really hard to improve our robot for the next year Robocup@Home competition. As this would be our fourth participation, we learned a lot at Montreal Robocup and we made many improvements to get better results, mostly on the software side. In the past, the team went in many competitions like the Eurobot but made the leap for the RoboCup@Home competition to get a bigger challenge and to get an opportunity to bring novelty in the scientific community surrounding robotic.

SARA, our creation, was designed for polyvalent human-robot interaction as well as efficient navigation and object manipulation. Our robot is mounted on four mecanum wheels powered by Roboteq drives, a 7 DoF arm and sensors for communication and navigation. Our team has developed knowledge in the object and people detection/recognition, as well as navigation using a laser scanner, odometry on the wheels and an Asus Xtion camera. All of these parts are interfaced through ROS (Robot Operating System).

In the rest of this paper, we will present in the 2nd section the mechanical improvements we've made to our robot to overcome the different challenge. In section 3, the different packages we've developed are described. And finally, this paper will conclude and explore the expected features for next year Robocup.

2 Mechanical improvement

To improve our robot abilities, we decide to add a vertical linear actuator, more specifically, a TL5 column made by TiMOTION. This will add a degree of freedom, giving us a wider range of motion to reach objects on the floor or higher on the cupboard shelves. This will be really helpful for a challenge like storing groceries where the objects can be anywhere in the cupboard.



(a) Robot retracted

(b) Robot extracted

Fig. 1: S.A.R.A. linear actuator motion range

We also decided to improve our wrist by adding a gearbox, giving it more strength. We took the decision to improve it after having problems with the gripper being too heavy.

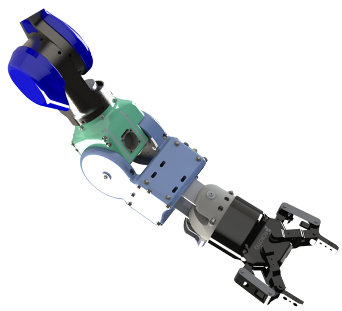


Fig. 2: Improved wrist 3D render

3 Software

3.1 Natural language understanding

To convert spoken data in actions subset, we had to create our own natural language understanding system (http://github.com/walkingmachine/wm_nlu. To do this, we based ourself on rasa nlu[1], an open-source NLP tool for intent classification and entity extraction. But a simple entity extraction wasn't enough for us, we wanted a system that would take a command as an input and output the desired actions.

To do this, our first step was to create a dataset for the entity classification. Based on the GPSRCmdGen, we generated sentences which we hand labeled by attributing an entity to each specific type of sentence paired with specifics parameters.

We then built a ROS service which takes a sentence as an input, classify the main intent using our dataset and return an array of actions that the robot needs to execute according to the command. Our system is dependent on our environment representation package(http://github.com/walkingmachine/wonderland) since queries are made to our database.

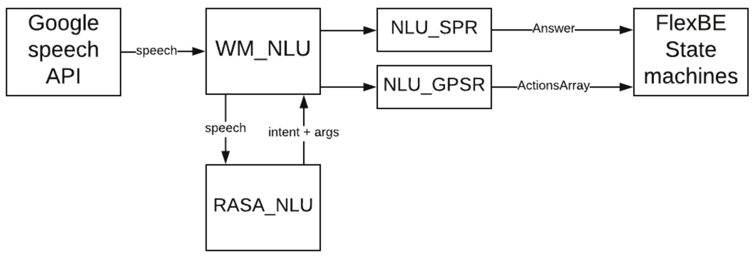


Fig. 3: Natural language understanding process

3.2 Sound localization

To improve our performances mainly in the SPR challenge and to add reactivity to our robot, we decided to add a Matrix Creator which include a microphone array coupled with a Raspberry Pi 3. We decided to use ODAS [?] which stands for Open embeddeD Audition System. This is a library dedicated to performing sound source localization, tracking, separation, and post-filtering, developed by IntRoLab[?] from Sherbrooke University in Quebec.

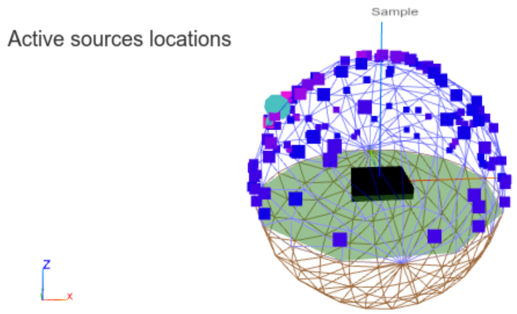


Fig. 4: sound source localization from ODAS web visualizer [?]

At first, a simple sound source localization is used. But then a Kalman filter is applied to perform sound source tracking. It helps us eliminated simple noise to tracking a person talking to the robot. We can even go further with this library by using the sound source separation which helps us separate the sound incoming from the different speakers.

We decided to build our own ROS wrapper around ODAS considering the lack of documentation surrounding the project. Our wrapper offers multiple topics which publish either the different sound sources localization, the tracked sound source or the separated sound sources. We can then easily identify the location of a speaker giving a command to our robot.

3.3 Object recognition

Recognition system

For our object recognition, we use YOLO [3], a real-time object detection. It does not only detect various object but it also predicts the bounding boxes of the detected object. It uses a single neural network which is applied to the image. Multiple regions are then created and are used to predict the bounding boxes. Each of them also contains the predicted probability which is used to filter the predicted objects. The advantage of this system is that it can detect multiple objects in a real-time scenario.

Dataset creation tool

This year we are putting our efforts on a way to simplify the dataset creation. During the last Robocup in Montreal, it was the first time our team had an efficient object recognition system. But our flaw was in the production of our dataset. Since we are retraining over ImageNet pre-trained weight, we need to

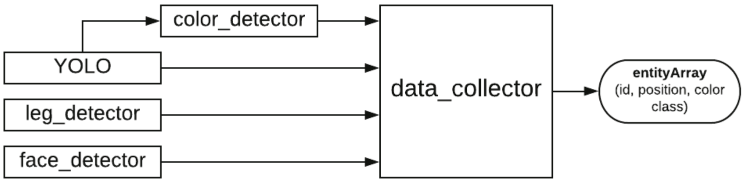


Fig. 5: CHANGER IMAGE POUR OBJECT RECOGNITION

provide a large dataset and for this, we had to do all the bounding boxes by hand for every image.

We decided that we needed to find a faster way to train the provided objects from the arena. Our plan is now to use a rotating platform with a green screen, that way we could automate the data collection process by using background subtraction technique with OpenCV and contour detection to find the object bounding box. Using the subtracted object, we can now apply different transformations to do dataset augmentation.

For the moment, we only created the software part to automated the bounding boxes. As you can see, on the Fig.4 a), we applied an InRange filter, on the Fig.4 b), we inverted the image to finally apply the findcontour function on Fig.4 c). By taking the largest contour found, we can easily calculate the bounding box of the object.

(a) InRange filter (b) Invert image (c) Contour detection

Fig. 6: Dataset creation process with wm_dataset_preparation

3.4 Environment modeling

To have a good representation of the robot environment, we develop our own environment modeling we called Wonderland [?]. For this, we can insert in our database all the known information like the furniture, the rooms, and the objects. Based on this, our robot will interact with the database through a web API, which makes it ROS agnostic and could be used by any application even if it's not robotics oriented.

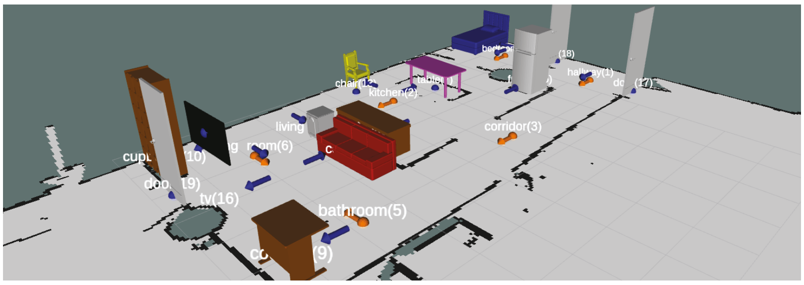


Fig. 7: wm_data_collector process

To make our life easier, we also implemented an easy way to place the object's position through Rviz. For the moment we use the pose estimation tool but we are planning to develop our own tool in Rviz next year.

With our environment modeling system, our NLU node can interact with it to confirm that a query is valid, by verifying by example the existence of a specific room or a specific object.

4 Objects and people tracking

To keep track of the many objects and people our robot sees, we have developed our tracking system named wm_data_collector [?]. Our process, as you can see on Fig8, imply the use of our different sensors to keep track of different characteristics defining our tracked entities. Once we gather the information, we use a simple Kalman filter but we are planning to use a better algorithm in the future like dlib correlation tracker.

5 Conclusions and future work

In this paper, we presented how we are developing our own robotic platform for the Robocup@Home competition. It has many abilities like a person and gender recognition, pose detection, environmental reasoning, object recognition, and manipulation and many more. Since we are mainly undergraduate students

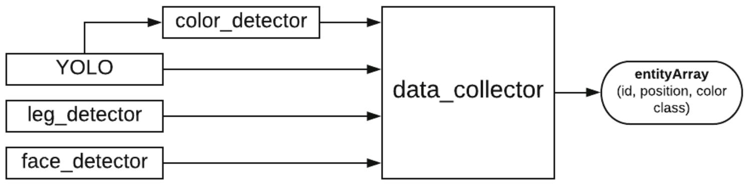


Fig. 8: wm_data_collector process

with no teacher support, our main efforts go in the learning process and the implementation part. Despite a low scientific contribution, we are actually developing some package that we hope will be usable by other teams in a near future.

This year we are planning to implement gesture recognition for the restaurant challenge and planar and object segmentation to detect unknown objects mainly for the storing groceries challenge. For this, we'll simply use PCL library with the RANSAC algorithm using a plane model and the K-mean algorithm for object segmentation. Finally, we'll also be working to improve our Natural Language Understanding package to add more features for the more complex challenges in the GPSR.

References

1. Rasa nlu: Language understanding for chatbots and ai assistants.
2. Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017.
3. Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. *arXiv preprint arXiv:1612.08242*, 2016.
4. Philipp Schillinger, Stefan Kohlbrecher, and Oskar von Stryk. Human-Robot Collaborative High-Level Control with an Application to Rescue Robotics. In *IEEE International Conference on Robotics and Automation*, Stockholm, Sweden, May 2016.
5. Bruno Siciliano and Oussama Khatib, editors. *Springer Handbook of Robotics*. Springer, 2016.
6. Roland Siegwart and Illah R. Nourbakhsh. *Introduction to Autonomous Mobile Robots*. Bradford Company, Scituate, MA, USA, 2004.
7. Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *CVPR*, 2017.
8. Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *CVPR*, 2016.
9. Stephen J. Wright. Coordinate descent algorithms. *Math. Program.*, 151(1):3–34, June 2015.

Robot SARA Hardware Description

Specifications for robot SARA are as follows:

SARA	
Base	Custom base with fully holonomic platform
Vertical column	Timotion TL5
Right arm	7 DoF custom arm made of Kinova motors and dynamixels
Neck	Tilt and pan unit using two Dynamixel MX-64R servo actuator
Head	Custom head made of RGB neopixels leds and Asus Xtion Pro
Gripper	Robotiq 2 fingers 140mm
Dimensions	Base : 0,61m. X 0,77m. Height : 1,48m.(min.) 1,78m.(max.)
Weight	∼70kg
Additional sensors	Hokuyo UTM-30LX on base
Microphone	Rode microphone
Sound localization	Raspberry Pi 3 and Matrix Creator
Batteries	2x 20V Dewalt drill battery 5aH
Computer	1x Lenovo p50 with 32GB RAM and nVidia Quadro M2000 4GB, 1x Raspberry Pi 3, 1x Nvidia Jetson TX2

Table 1: Robot's hardware description

Robot's Software Description

For our robot we are using the following software:

- Platform: Robotic Operating System (ROS) Kinetic on Ubuntu 16.04
- Navigation, localization and mapping: Gmapping, AMCL, pointcloud_to_laserscan
- Face recognition: ros_face_recognition
- Speech recognition: Google Speech API
- Speech comprehension: wm_nlu
- Sound localization: wm_odas_ros_wrapper
- Speech generation: Svoxpico,MaryTTS
- Object recognition: Darknet with YOLO v2
- Arm control: MoveIt and Kinova API
- Task executor: Flexbe
- World reprensentation: Wonderland

Fig. 9: Robot SARA

Team members

André-Philippe Audette	Electrical engineering bachelor student
Nicolas Bernatchez	Manufacturing engineering bachelor student
Jeffrey Cousineau	Manufacturing engineering bachelor student
Raphael Duchaine	Software engineering bachelor student
Quentin Gaillot	Manufacturing engineering master student
Louis-Charle Labarre	Manufacturing engineering bachelor student
Philippe La Madeleine	Manufacturing engineering bachelor student
Redouane Laref	Manufacturing engineering bachelor student
Huynh-Anh Le	Manufacturing engineering bachelor student
Lucas Maurice	Software engineering bachelor student
Alexandre Mongrain	Manufacturing engineering bachelor student
Jimmy Poirier	Electrical engineering bachelor student
Veronica Romero Rosales	Manufacturing engineering bachelor student