CATIE Robotics @Home 2024 Team Description Paper

Sébastien DELPEUCH, Pierre-Marie ANCÈLE, Thierry ARSICAUD, Alban CHAUVEL, Charles DORMOY, Christine JAUREGUIBERRY, Clément LAIGLE, Clément PINET, Jean-Noël BARTHAS, Florian LARRUE, and Sébastien LOTY

Centre Aquitain des Technologies de l'Information et Électroniques (CATIE)
1 Avenue du Dr Albert Schweitzer, 33400 Talence, France
`s.delpeuch@catie.fr`
`https://robotics.catie.fr/`

Abstract. This paper provides an overview of the CATIE Robotics Team's activities for participating at the RoboCup @Home 2024 in Eindhoven. A TIAGo[1] called Epock is used as the main robotic platform. Safe and autonomous navigation is achieved by integrating ROS compatible components. Totally offline, speech recognition and natural language understanding are performed by Whisper model and an open source NLU from SNIPS. State-of-the-art neural networks are used for face recognition, person tracking and object detection. Problem-specific developments are successfully tested for object localization. A human-centered approach is used to enhance Epock's interactions. An environment perception-based grasping pipeline using octomaps is implemented. This document describes selected components, shares some feedback and discusses possible improvements.

1 Introduction

The CATIE Robotics Team was formed at the beginning of 2018 and is part of CATIE, a RTO (Research and Technology Organization). CATIE is a non-profit organization supported by the Nouvelle-Aquitaine French region whose mission is to assist companies willing to adopt and integrate digital technologies in their technological and economic development. By creating a RoboCup@Home team, our ambition is to be part of an expert community, nurture our robotic knowledge and share it to foster progress towards a tangible goal. The competition offers an objective benchmark to measure progress and is a stimulating means to unite efforts locally.

In 2018, we focused on integrating proven technologies and achieving simple but robust behaviors in key fields such as safe navigation, Simultaneous Localization And Mapping (SLAM), force-based grasping, person and object

[1] `https://tiago.pal-robotics.com/`

recognition. TIAGo, an off-the-shelf service robot, was chosen as our main development platform and Robot Operating System (ROS) as our middleware framework. This approach worked well in our first international competitions: we ranked second in GermanOpen 2019 RoboCup@Home[2], third in Sydney 2019 RoboCup@Home, second in SciRoc Episode 7 challenge and third in Bordeaux 2023 Robocup@Home.

Bordeaux 2023 helped us to achieve high-level, integrated, robust, modular and safe behavior for our robots. In this last year, the MoveIt! based manipulation pipeline was improved, together with the servo-control of the mobile base to enable local control of the robot. It also facilitated the integration of new artificial intelligences, simplifying the addition of complex behaviors, etc.

Now that we're entering our 5th year, our aim is to achieve high-level, complex behavior, focusing on system reliability and robustness while maintaining a high degree of modularity. This implies addressing more complex tests such as GPSR, requiring a platform with a high awareness of its environment and context.

In this paper, developments related to the navigation, perception, communication and object manipulation capabilities of Epock, as well as hardware modifications are presented. For each section, an overview of the approach, the results and the future work is given.

2 Navigation and SLAM capabilities

A robust and reliable navigation has always been our priority. We have designed a complete and functional pipeline from map creation to autonomous navigation and localization. No collisions occurred during the overall long distance covered by the robot in different environments throughout the last year We use AMCL for localization and ROS move_base for navigation. Furthermore, we have added remote encoders to the robot, allowing us to calculate the robot odometry which will not be impacted by wheel skidding. It enables us to improve our localization through a Kalman filter. Moreover, we are working, on the local control of the robot through the implementation of a mobile base control. This allows us to locally deal with problems for which the ROS native navigation stack is not sufficient, such as going through a door or moving in a confined area. As of this year, we are working on improving our various planners with a focus on adding social filters to the robot. This aims to facilitate smoother and more human-like navigations.

3 Perception capabilities

Perception problems are mainly tackled by several state-of-the-art neural networks, detailed below. For each neural network, we have tried to find the best compromise between accuracy, performance and resource consumption.

[2] https://www.youtube.com/watch?v=7Y4RjxWRqxE&t=5s

Moreover, we emphasized the reusability of our algorithms, minimizing their dependency on a framework. Consequently, all the AIs presented below are embedded in an environment called Eagle Nest. This environment provides a uniform AI catalog that can be used in various projects. In addition to standardization, it offers a monitoring and execution authority for algorithms, ehancing errors and bugs management. This not only reduces the integrating costs of new AI but also increases maintainability and robustness, simplifying the creation of robust complex behavior.

We mostly rely on siamese networks. This type of neural network can learn a similarity measure between objects [**?**]. Every object is mapped by the network to a vector. Object comparison can then be achieved using the distance between vectors. This enables these networks to naturally generalize to unknown instances. We also sometimes use some more problem specific approaches, described in dedicated subsections.

3.1 Object Recognition

Object recognition is mainly done using YOLO V8, that was recently integrated in our Eagle Nest environment to enhancing the processes and performance of the object recognition algorithms. Data sets used during the last RoboCup@Home were thus generated in the YOLO format, by aggregation of 3D reconstructed views of YCB objets, and a serie of annotated pictures extracted from videos taken in situ during the setup days, using a moving camera and a rotating podium. Particular attention was paid to ensure quick and reliable labelization of targeted objects, on site, using dedicated manual labeling tools and scripts combined with the use of the Segment Anything tools. Using YoloV8 with this process allowed us to speed up shooting, labeling and training - compared to our previous approach, mainly based on MaskRCNN - so that we were able to carry out re-trainings on site for specific events, in a matter of minutes. When needed in our algorithms, segmentation was performed, on top of object detection made by Yolo, with the help of the Segment Anything model (”bounding box prompt mode”). This might be replaced, in the future, by YoloV8 segmentation, which will give us better control on specific targeted objects and shorter inference times for these tasks. Other machine learning models may also, in the future, be tested and used to improve the performance of recognition algorithms, in terms of success rate, use of onboard resources and /or inference time, depending of the context. But pragmatic approaches sometimes work better than neural networks , as we demonstrated in previous editions. In the SciRoc Pick and Pack challenge, we used point cloud depth filtering to detect objects, based on the fact that they are on the shelves. We also took advantage of objects not being too close to each other to identify distinct blob of points for each object, with basic color filtering. This code was written with OpenCV. Beyond object detection techniques based on 2D image processing, we also plan to test / explore approaches based on RGBD image and point clouds processing (3D feature extraction, 3D CNN...).

3.2 Face detection and recognition

Face detection and recognition is done by a combination of two neural networks. The first one is a network that is used out of the box and that performs the face detection using a state-of-the-art Multi-Task Cascaded Convolutional Neural Network (MTCNN). Once the face bounding boxes are extracted as well as face features (eyes, nose, ...) , the image is cropped and normalized. Faces are then processed through a siamese network. This network has been trained on more than 8000 different identities using the publicly available VGG 2 dataset. The distance between the faces and the previously detected people are computed and the closest person according to the L2 distance is returned if the value is below a predefined threshold. Examples of uses include checking if the right person is following Epock or during the *Find my mates* task in RoboCup@Home for identifying people faces.

3.3 Person attributes detection

We trained a neural network to identify a person's attributes from a picture.

This network was trained on the CUHK-PEDES dataset using a resnet architecture and image-text pairs composed of sentences discribing hair color, clothes, accessories... of people shown on the corresponding picture.

The generated embeddings database is later searched, for a given person picture, in order to generate it's description.

This model, an algorithm, is one of the core features used in the Receptionnits task in the RoboCup@Home.

3.4 Person tracking and *follow me* behavior

During the *follow me* phase, we control the robot's head to always be looking in the direction of the target's last known position. Using this position as target position and depending on the situation, we use either

- Dijkstra's algorithm to create the path to the target by taking into account the obstacles recorded on the cost map by the LIDAR and the RGB-D camera. The velocity commands are calculated by the local planner.
- the robot local servoing of the robot allowing it to evolve in a more restricted environment.

3.5 Pose Recognition

The pose recognition is done with the Movenet library[3] which computes face, body and hands 2D keypoint detection in real-time if equipped with a recent graphics card. Movenet gives the Body-Foot Estimation (default skeleton), but in the @Home competition, we need a more high-level information like "the

[3] `https://www.tensorflow.org/hub/tutorials/movenet?hl=fr`

person is seated" or "the person is pointing to the left". This is determined by simple rules based on the skeleton, such as: "If the knees are approximately at the same height than the hips and the neck is significantly higher, then the person is sitting". We wrote this code on top of Movenet. It can categorize the person's pose (sitting, standing, raising arm, etc.). This algorithm is our gateway for everything related to individuals. As its resource consumption is very low, it is used to prevent other algorithms from spinning in a void. Typically in Receptionist, where we need to re -identify the host or first guest, Movenet is used to detect people and automatically filter out those outside the arena. It is also used to center people within the image to perform re-identification or description of people, which are more resource-intensive and need to be minimized.

3.6 Precise object localization

When Epock has to interact with an object at a known position in the environment, it cannot only rely on its own localization.

Sometimes the actual object position can be slightly different from the theoretical one and Epock's localization alone is not accurate enough for precision tasks such as grasping.

For example, in the *take out the garbage* challenge from RoboCup@Home, robots need to detect garbage bins to take trash bags out of them. In *SciRoc Pick and Pack* challenge, robots need to drop the items in a crate. These tasks require a precise localization of both the bins and the crate.

A LIDAR based pattern recognition is used to detect objects with a specific geometry. Indeed, a trash bin seen by the LIDAR is a semicircle. To add more robustness to this detection, the object theoretical position is also taken into account. An object matching the geometry, but located far from its expected position is probably not the seeked one.

This approach is coupled with our work on local control of the mobile base, enabling us to precisely reposition the mobile base using LIDAR data.

4 Communication capabilities

For speech-to-text and natural language understanding (NLU) capabilities, CATIE has developed a pipeline using OpenAI's Whisper model and an open source NLU from the SNIPS project. Whisper [4] is an automatic speech recognition (ASR) system trained on 680 thousand hours of multilingual and multitask supervised data collected from the web. Snips-NLU [5] is a free, ready-to-use solution for non -commercial software to select the most relevant transcripts. Moreover, a node for assembling these two text models has been developed. The result is an NLP solution with offline operation and low latency, which is essential for the RoboCup competition.

[4] `https://openai.com/research/whisper`

[5] `https://github.com/snipsco/snips-nlu`

During previous competitions, we noticed several sources of error and variability. First of all, a notable difference was the amount of noise during the competition, which was corrected by the purchase of a quality directional microphone. In addition, we noticed that our pipeline tended to over-interpret: even without a speaker, it detected sentences in the white noise, which were then interpreted by the NLU. To address this issue , we correlated the activation of the microphone with specific robot states. So, instead of listening continuously to the environment, it only listens when it receives to extract from its environment.

For text-to-speech, we are using TIAGo's default text-to-speech module: Acapela[6].

The different challenges encountered during our competitions were an opportunity to see how Epock interacts with people. We noticed that some people had difficulties understanding when to speak to Epock or what Epock was doing . To address this issue, we worked on interactions with a Human-centered approach by adding a screen where the current state of Epock is displayed (help needed, listening, moving, in action). Moreover, the integration of different leds make possible to know that the robot is performing actions (arm movement, waiting for speech, etc.). A user study was conducted to see if these pictograms were well understood and to get some feedback. Impressions and comments of people not involved in the project were collected. Now, with the additional pictograms, users can adapt their behavior according to the robot state shown on the screen as they would with another human. Our interface is based on Bastien and Scapin's Criteria.

Futermore , we provided light and sound signals to indicate when the robot's readiness to listen, when its lack of understanding, and so on. Finally, for the acquisition of complex instructions, such as placing an order in a restaurant, we designed an activity diagram to enable the robot to ask the appropriate questions and request the appropriate confirmations. This enables us to collect information quickly, while maintaining the quality of the user's interaction and concentration.

5 Grasping capabilities

5.1 Arm control

Since the 2020 TDP, most of our effort has been focused on implementing a prehensile pipeline. The aim is to implement a succession of actions allowing the robot to : locate an item in space, create a representation of this space (octomap) and plan a trajectory in this high dimensional space using MoveIt!. Without forgetting our safety and robustness approach. This grasping pipeline is therefore an element of a more general grasping core which, depending on the situation, decides to use one module or the other of the grasping pipeline (octomap generation or not, etc.).

[6] http://www.acapela-group.com/

To reach an acceptable security level, we use high level guidelines and low level behaviors. The guidelines include:

- Position the robot in order to maximize the surrounding space.
- Only move the arm if the robot is still.
- After grasping an item, try to retract the arm as best as possible before moving the base of the robot.
- Always announce vocally that the arm will move.

Epock's force sensing capabilities are also used to improve the overall robustness of the grasping pipeline. For example, if the weight measured by the wrist sensor goes below a predetermined threshold, the robot will assume that the item fell off its gripper and try to grasp it again if possible[7]. Also, some items are smaller than the precision we're capable of reaching at the end of the gripper along the Z axis (e.g. a fork on a table). In this case, we'll use a slow descending motion until resistance is felt when the table is touched. The same approach was used to grasp the garbage whose size is unknown prior to the test.

We've tested the impact of using checkpoints to achieve a reproducible behavior and tried to evaluate the appropriate force to apply on objects depending on their resistance and stiffness. Formalizing these methods and making a generic module out of them is a work in progress.

Finally, we are currently focusing on the robot's ability to perform the manipulation in all cases. Rather than improving the quality of gripping algorithms as much as possible, as it falls outside our areas of expertise , we prefer to adopt a more robust approach. Implement a fully controlled grasping pipeline where the failure of one part of the pipeline does not jeopardize the failure of the entire grasping process. This involves systems for checking collisions, gripping success, etc., to detect abnormal states and define corrective actions while maintaining a safe state. This work enables us to have a state-conscious robot, and thus to choose the level of risk to be taken according to a given state (moving quickly towards the objective, choosing a complex trajectory, asking for help, etc.).

5.2 Perception challenges

Grasping requires specific information from the item. Such as its position, mask or orientation. We reviewed the state of the art on object recognition for grasping in 2020. Further to this state of the art, the implementation of object detection strategies based on several methods has been performed. On one hand, a MaskR-CNN algorithm has been developed and enables to provide the bounding box and the mask of the different objects. This may be replaced in future work by YoloV8 segmentation, which will allow us to accelerate the processes and treatments on labeling, training and inference. On the other hand, the specific characteristics of the objects (color, height, ...) are identified thanks to custom OpenCV code.

[7] `https://www.youtube.com/watch?v=btS7S6dadN4`

Using the latter, we obtained good results in the final task *Pick and Pack* of the SciRoc 2019 competition (6/6 objects chosen, 5/6 objects packed).

Work on point cloud analysis has been recently undertaken. This allows, among other things, to detect flat surfaces (such as a table) where the robot could drop an object. Performing object detection using the point cloud rather than the 2D color image is then possible.

6 RoboCup experience and community outreach

In addition to the results presented in the introduction, we were exhibitors at 2018 and 2019 Cap Sciences' Village des Sciences, that gathered more than 3000 people over the weekend around robotics and the RoboCup competition [8]. We took part in the following events in 2019: NAIA (Bordeaux) and Vivatech (Paris). We participated in RoboCup@Home Education Challenge @EURCJ 2018 and co-organized a similar workshop in early 2019 that gathered 30 students in Bordeaux.

We carried out demonstrations of the RoboCup tests at NAIA.R (Bordeaux) in 2021 and during the SIDO exhibition (Paris) in 2021 and 2022. In 2023 we collaborated to promote national and regional events like Robot Maker Days or RoboCup Junior in Bordeaux.

7 Conclusion

In this paper, we have given an overview of the approaches used by the CATIE Robotics team for the RoboCup@Home competition. We have detailed our approaches for navigation, detection, communication and grasping. In all these areas, we have made significant improvements, but we are still building a robust basis for the competition by catching up to the state of the art, which consumes most of our time.

[8] http://www.cap-sciences.net/au-programme/evenement/village-des-sciences-2018

Epock - Robot TIAGo Hardware Description

Robot TIAGo has been selected and is being customized for the @home competition purpose. Specifications are as follows:

- Base: differential drive base, 1m/s max speed.
- Torso: lifting torse (35cm lift stroke)
- One arm with a gripper (7 DoF). Maximum load: 2kg.
- Head: 2DoF (pan and tilt)
- Robot dimensions: height: 1.10m - 1.45m, base footprint: 54cm diameter
- Robot weight: 72kg.

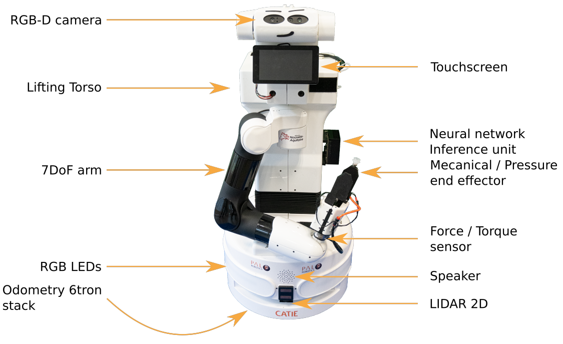

Fig. 1. Robot TIAGo

Our robot incorporates the following sensors:

- 2 RGB-D camera
- 2D LIDAR
- Stereo microphone
- Speaker
- 3 Sonars
- 2 IMU
- Motors current feedback
- Wrist force and torque sensor
- Pump pressure sensor

Robot's Software Description

For our robot we are using the following software:

- OS: Ubuntu 18.04
- Middleware: ROS Melodic
- Simulation: Gazebo
  http://gazebosim.org/
- Visualisation: RViz
  http://wiki.ros.org/rviz
- Localization: AMCL
  http://wiki.ros.org/amcl
- SLAM: Cartographer and GMapping
  https://github.com/googlecartographer/cartographer
  http://wiki.ros.org/gmapping
- Navigation: move_base
  http://wiki.ros.org/move_base

Robot software and hardware specification sheet

- Arms control: moveIt! and play_motion
  http://moveit.ros.org/
  http://wiki.ros.org/play_motion
- Face recognition: custom siamese neural network
- Object recognition: YoloV8 neural network and custom solution
- Pose detection: MoveNet
  https://www.tensorflow.org/hub/tutorials/movenet
- Person Re-identification: based on Market 1501
  https://paperswithcode .com/task/person-re-identification
- Speech recognition: custom solution based on Whisper and SNIPS (see dedicated section)
- Speech generation: Acapela
  http://www.acapela-group.com/
- Task executor: SMACH
  http://wiki.ros.org/smach

External Devices

Our robot relies on the following external hardware:

- Rode Videomic Pro external microphone
- External laptop
- 1 touch screens
- 2 6TRON stack developped by CATIE
  https://6tron.io/
- External GPU NVIDIA Jetson Orin