Team Description Paper of TJArk@Home

He Zongtao
Robot and Artificial Intelligence Lab
Tongji University
1930719@tongji.edu.cn

Zhou Xun
Robot and Artificial Intelligence Lab
Tongji University
1930722@tongji.edu.cn

Xu Weihan
Robot and Artificial Intelligence Lab
Tongji University
1552360@tongji.edu.cn

Abstract

RoboCup@Home is one of the world largest robotics competition aims at bringing robotic platforms to use in realistic domestic environments. Although the competition has been held for years and many teams contributed a lot, there are still obstacles in service robotics technology, such as precise grasping, real-time object detection, etc. As to Social Standard Platform League with the Pepper platform, some tasks become more difficult due to limited hardware performance. In this paper, we describe our team's work on this competition and outline functions we have implemented on Pepper. Our software framework is homemade, including high-level package of native NAOqi API, ROS-based components, application skills and HTTP communication mechanism. We develop visual-SLAM and real-time naviagtion modules based on ROS and RTAB-Map. Cameras and deep learning are used to deal with environmental reasoning and interaction tasks. There are also many interesting applications, such as conversation, action imitation. Finally, we designed a user-friendly GUI on Pepper's pad, so that any amateur can control the robot and enjoy its service. When we first took part in China RoboCup@Home competition in 2019, we won the first-place. We are eager to do more and show more on a broader stage.

1 Introduction

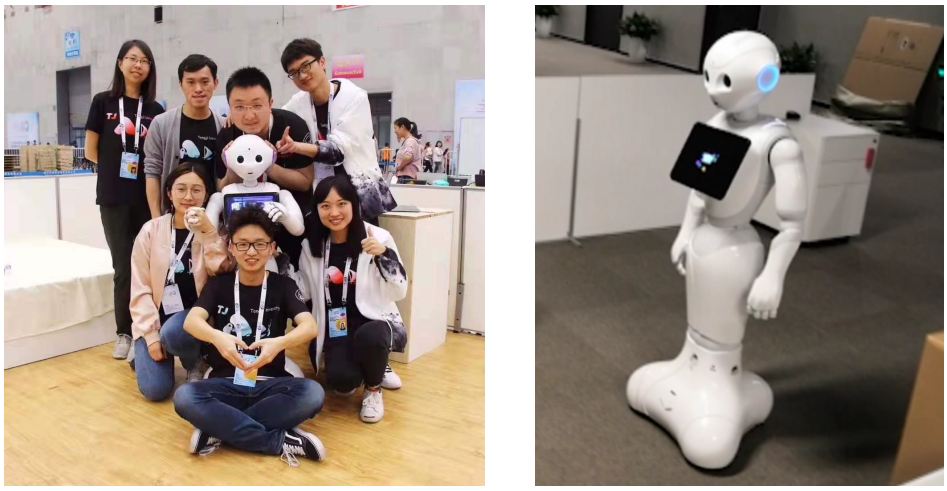
As development of robotic technology, more and more people expect intelligent robots to serve the family as human beings and provide a good and comfortable life experience. The RoboCup@Home competition [1] focus on bringing robotics to apply in common domestic environment. To achieve such goal, the robotic system must be able to perceive environment, communicate with humans and complete various tasks, such as cleaning up a table, holding a party or guiding new comers. Security is the top priority. Because service robots can't do harm to people and the environment. Secondly, the robot should be controllable. The operator can intervene the behavior of the robot at any time to prevent the unexpected situation. Finally, the robot should have complete autonomous action ability and give people comfortable interaction experience.

The Robotics and Artificial Intelligence Lab (RAIL) of tongji university was founded in 1992, Team TJArk of RAIL was founded in 2004 and participated in RoboCup World Cup from 2006 to 2018. We have got seven winning streak in China RoboCup SPL and once won the third place in RoboCup2018 SPL. In December 2018, we founded an energetic new team to participate in RoboCup @Home League, called TJArk@Home. Main goal of TJArk@Home is to explore the

limit of how well can robots serve people during common life with acceptable cost. Under the guidance of such wish, we are doing many research on the Pepper platform, such as

- SLAM(Simultaneous Localizaiton and Mapping)
- Auto navigation of robot
- Object Detection and Recognition
- Human-face detection and analysis
- Human-Robot interaction
- Motion control
- Trajectory teaching
- Design of laptop GUI
- ...

April 2019, the first time TJArk@Home involved in China RoboCup@Home League, we got the first-place in SSPL(Social Standard Platform League). During the competion, we show a lot of abilities, including Autonomous navigation, Follow an operator to the given position, Recognize speech and response, Detect humans in vision and summarize their features. Although our team



(a) Team photo at China RoboCup 2019 (b) Pepper in homemade scenario

Figure 1: Our Pepper

has not been established for a long time, we have developed a reusable and extensible software framework. It is a higher level integration of NAOqi API, which makes developing a new application more easily. ROS-based components with a bridge to NAOqi are also developed. The HTTP-based Information Exchange System(HIES) also helps a lot. Tasks require large computational resources such as syntactic analysis can be allocated to a host server. Pepper just need seed raw data and listen for response. Such mechanism reduce Pepper's burden, allowing it pay more attention to interaction logic and task excution.

Tongji RAIL lab has been researching on robotics since its begining, and achieved a lot in several aspects, such as humanoid walking [2, 3], object detection [4], fuzzy systems [5]. RoboCup@Home League is a new attempt, which involves some former research, but bringing untouched challenges like human-robot interaction. We are pushing researchs on vision, navigation, interaction and building a simulated competition scenario for integration testing.

2 Architecture

As shown in Figure 2, we have developed a Python framework, which is a high-level package of NAOqi APIs and ROS-based components.

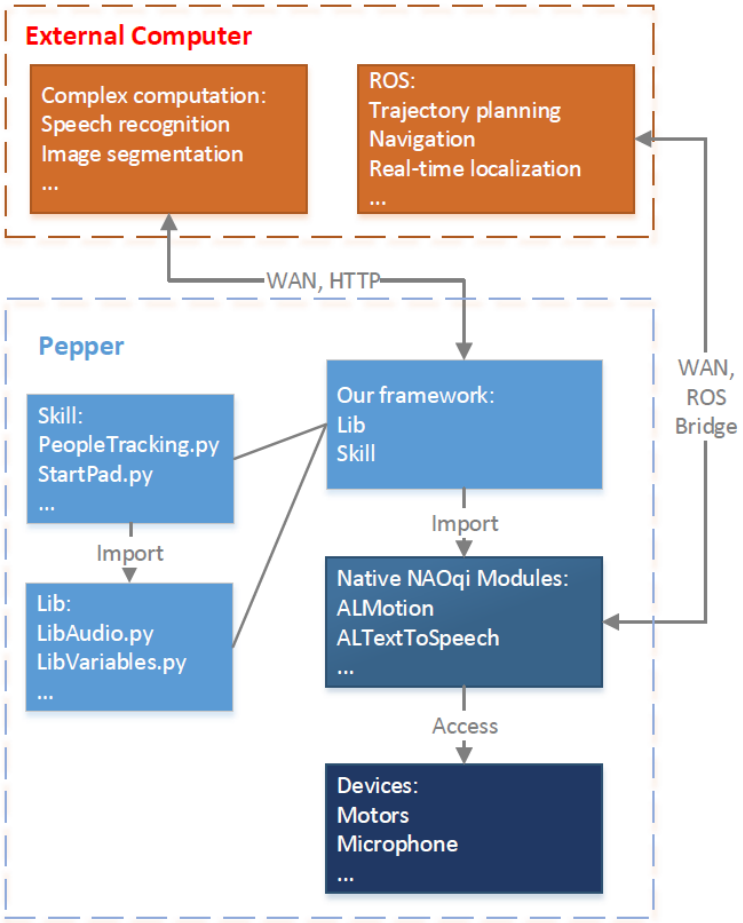


Figure 2: System architecture

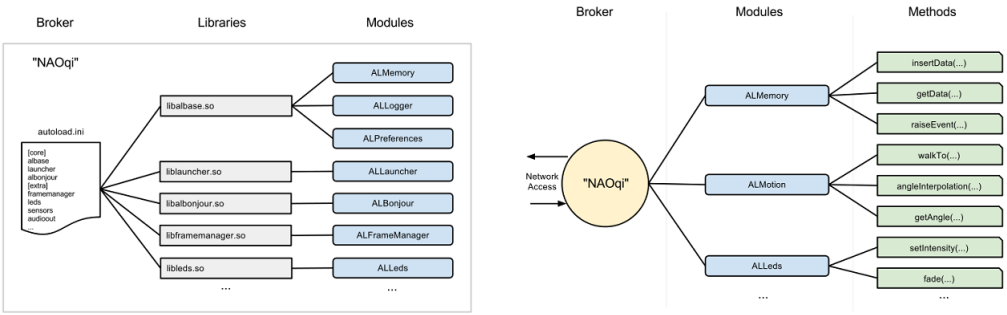
2.1 NAOqi Operation System

In most aspects, NAOqi OS behaves like a Linux. The most significant feature is its NAOqi modules. Various modules can be accessed through network by the proxy called "ALBroker". To upper level application, the proxy is transparent, which makes remote access has no difference with local access. All underlying hardware can be used by these NAOqi modules easily. Figure 3 shows the NAOqi architecture.

2.2 Homemade framework on Pepper

On NAOqi OS, several processes are simultaneously running, including UI process, HTTP server/client process, motion control process, vision process, navigation process, etc. All codes are organized into two main parts: libraries and skills. Libraries, we name them with "/Lib.*ṗy/" pattern, are files that contain various class definitions for basic ablilities. Every "Lib" attaches to some specific functions. For example, "LibVariables" defines a class related to global variables stored on local disk or remote server. "Lib"s are just useful tools without any task-oriented purpose.

To combine them together and implement specific applications, we write many app files, called "Skill". A "Skill" imports several "Lib"s, to accomplish a complete task, such as tracking a person. Every "Skill" runs as an individual process, so communication problem follows. OS's memory manager allocate different storage space for different processes, so there is no native global space for our different "Skill" and "Lib". But most of time, there is a large need for them to share information. To overcome such dilemma, we developed a JSON-based global variable mechanism. As mentioned



(a) File organization (b) Access process

Figure 3: NAOqi API

before, this function is implemented in "LibVariables". It uses hard-disk to store a global variable with JSON format. Thanks to separation principle of interface and implementation, the global space can be not only Pepper local disk, but also remote server memory/disk.

Overall, though many programs engaged, we just need to run a single shell script to start all applications we need. This native software system on Pepper is easy to learn, easy to use and easy to extend. Every team member can write his/her own lib files, such as "LibMoiton", "LibDetection", and add it to the framework by pushing their files to Pepper. The development process seems like building. It starts from a basic frame, and grows with bricks added by team members, and finally become a useful building.

2.3 HTTP-based Information Exchange System

Computational resources are limited in NAOqi OS. But in real applications quite a lot tasks need heavy computation, which require us to push them to a powerful remote computer. So, we designed a HTTP-based information exchange system. There are two HTTP server process, one in Pepper, another in external computer.

Main function of external computer server process is to listen to Pepper's requests, start corresponding computing program and response the result to Pepper. A conspicuous example is speech recognition task. After recording a complex speech sentence from the operator, Pepper sends the audio file to the external computer through HTTP request. The computer receives this request, and converts this speech to text. Finally, speech text is sent back carried by HTTP response package.

On the other hand, HTTP server process runs on Pepper mainly aims at web service. It contains a web site backend. When a browser login the URL, it gets a home page of various applications and can start them by simple clicks. Moreover, a device just need the ability of sending HTTP requests to control Pepper's behavior. This part will be described at ?? in detail.

The advantage of HTTP information exchange is obvious, it is widely used and device/language independent. But network quality would limit the performance of HTTP communication. Future works contain compressing information to reduce transfer load.

2.4 Robot Operation System (ROS)

Implementing SLAM and auto-navigation of robot needs well fusion of multi-sensor data and feedback control of motion actuators, so Robot Operating System (ROS) is a useful tool. We develop a basic module serving as a bridge between NAOqi and ROS, it publish Pepper's sensor data in ROS format and enable ROS to call NAOqi's APIs. ROS components get multi-sensor data, then do SLAM and path planning with real-time obstacle avoidance. As a result, motion commands will be sent to NAOqi.

3 Visual Simultaneous Localization and Mapping (VSLAM)

Pepper needs a model (a map) of the environment to support other tasks, mainly auto-navigation, but a prior map is often not available in various home scenarios, so SLAM find its application. The stereo camera data is mainly used by a visual SLAM module, together with laser (very limited) and odom (has significant error) data. Our VSLAM algorithm is based on RTAB-Map [6, 7], a graph-optimized based VSLAM algorithm, and have been modified for Pepper's sensors. Odom and laser scan data make the initial pose estimation, then stereo data serves for appearance-based loop closure detection to optimize local and global pose. VSLAM can get a 3D point cloud, we make a projection to get a 2D grid map for navigation. Having map and Pepper's size parameters, we use DWA (Dynamic Window Apporach) to plan the path and fuse vision, laser and sonar data to make real-time obstacle avoidance (mainly for dynamic scene), a series of motion command will be sent to NAOqi APIs, so Pepper can navigate to its goal points itself.

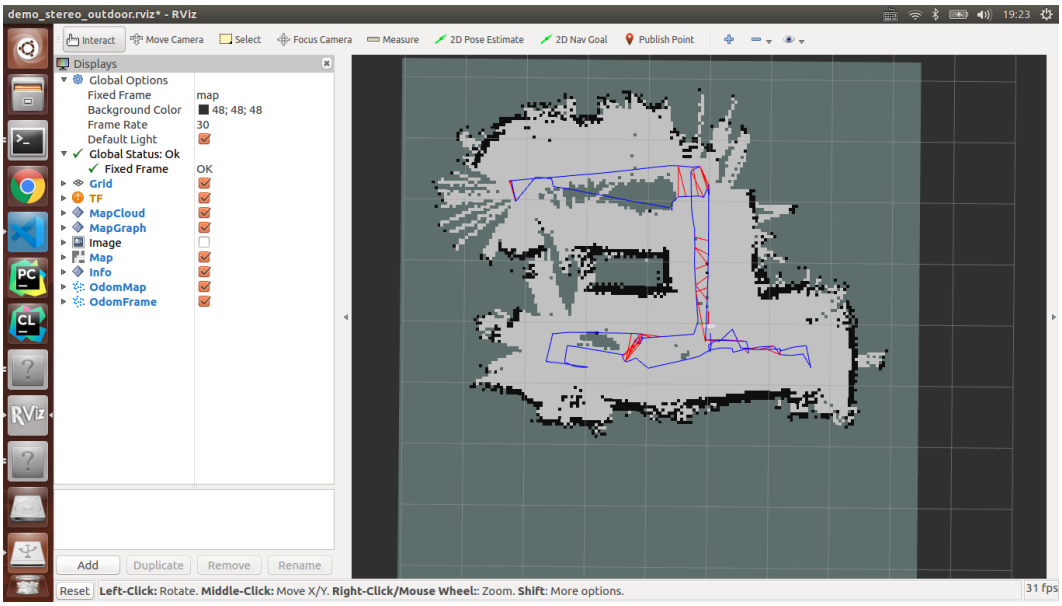


Figure 4: SLAM and navigation on ROS

4 Computer Vision

Various vision algorithms are needed for more careful perception and analysis of Pepper's environment on object and human level. Usually, to start a certain task, our Pepper can detect and recognize different people, remember their names and some other features, and tracking them. Then, more skills will be shown for complex goals. For object-level environment reasoning, we build an object detection and recognition framework based on YOLO [8], which is an interesting real-time algorithm. According to the competition scenario of RoboCup@Home, our object detection system can detect 50 classes of indoor objects in real time. To train our object detection network, we make a great dataset from OpenImages. We also match adjacent input frames to increase the robust of detection.

In some tasks we need a wide field of view, but the top 2D camera makes us disappointed. So we need to take some measures to expand Pepper's original "eyes". The method we choose is to stitch images from different perspectives, and a fusion algorithm follows to decrease the influence of illumination and geometric change. Now more visual candidates can be included for further processing.

A more exciting visual module is human pose estimation. For this feature, we use OpenPose [9] model to get a robust and real-time performance. With a non-parametric representation, we can detect associate body parts of multiple people in the input image stream.

Figure 5: Image Stitch and Fusion to Detect in a Lager Field of View

Figure 6: Using OpenPose model to estimate body pose

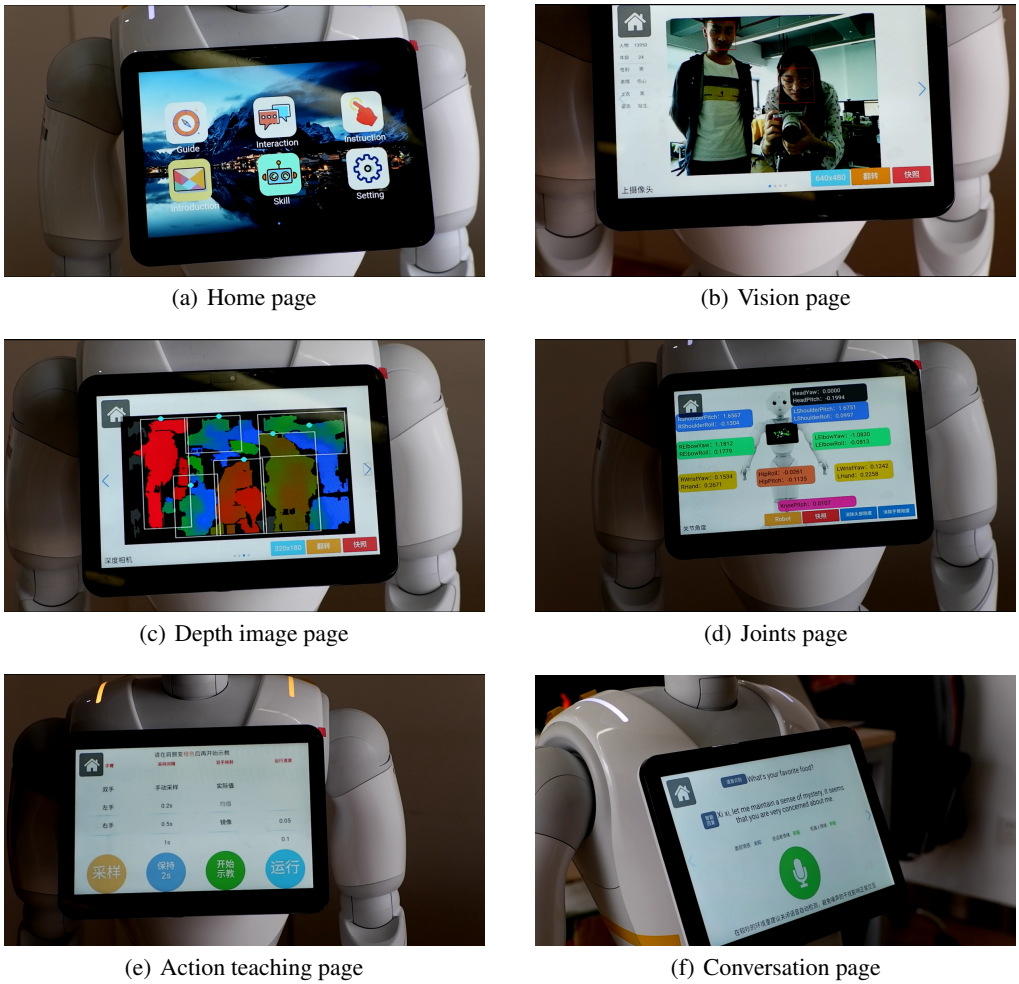
The NAOqi system itself contains many interesting APIs, they may not have high precision, but are still useful for further development. For example, we can use some APIs to judge whether a person is standing or sitting, close or far away from the robot. These functions are also integrated to human-robot interaction modules.

5 Human-Robot Interaction

5.1 Graphics User Interface

As a domestic service robot, smooth GUI interaction is neccesary. One of the advantages of Pepper is that it has a laptop on its breast, which makes GUI possible. We designed a user-friendly and cross-platform interface so even people unfamiliar to robotics can quickly operate on Pepper.

We have put twenty applications on the GUI so far, including Motion, Conversation, Vision, Joints, etc. In Motion, the operator can make the robot move a given distance to a given direction, or just move by a constant velocity. In Conversation, the operator can chat with Pepper using English or Chinese. In Vision, camera information is integrated. All images Pepper captured will be displayed here and the resolution is modifiable. In Joints, all joints' status are listed for developer to debug. We wish such GUI can help Pepper more powerful and useful.



(a) Home page

(b) Vision page

(c) Depth image page

(d) Joints page

(e) Action teaching page

(f) Conversation page

Figure 7: GUI on Pepper's pad

5.2 Speech Interaction

As mentioned in 5.1, we develop a conversation function. Language is an important interaction way in daily life. As a domestic service robot, it should have the ability to understand natural language. Our conversation applicaiton is powered by Xunfei cloud speech engine. The demonstration video has been uploaded to https://www.youtube.com/channel/UCE5ceR_uX-Hz3IIJx70lOrQ. However, this function is triggered by pad touch, which makes the chat not so less smooth. We are trying to make Pepper can detect and process speech of interest in real time.

5.3 Touch Interaction

To make Pepper look smarter, we develop some other interaction ways to enrich its behaviors. When someone touches its head, it would say something like "Huh, I feel a warm hand on my head". And if Pepper's hand is being grabed, it will softly close hand and hold you, and greet you. What's more, whenever pad or tactile sensors are touched, Pepper will look at that direction and speak or perform some body languages corresponding the condition. Such touch interaction is funny and important. It shows the robot is perceiving environment and ready to work.

For us developer, tactile sensors can do more. There are three parallel tactile sensors on Pepper's head. We designed a state machine to judge the touch direction. If the touch is front to back, the head stiffness will be toggled. If the touch is back to front, the arm stiffness will be toggled. If the hand is touched more than 3 seconds, corresponding arm stiffness wil be toggled. When we want

to change some stiffness, we just need touch the robot rather than back to computer and run some codes.

6 Conclusion

In this paper, we describe TJArk@Home's work towards RoboCup@Home competition and outline functions we have implemented on Pepper. After continuous improvement, our software has become a complete system, with many integrated applications. Besides software development, several researches are also embeded in our preparation for the competition, including computer vision, VSLAM, HRI.

Future works are wide-ranging. First, the software framework should be simplified and documented in order to increase re-usability and open source. Then, many application can't reach the real-time need for domestic tasks. So we will try to improve the efficiency and speed up program execution. A very tough obstacle on Pepper is that its fingers, with only one degree of freedom, are not flexible to finish accurate grabbing tasks. We are considering alternative solutions such as using two hands for grasping. Although a framework is developed, there are many specific tasks requiring highly integration of basic functions. So there is a long way for us to make Pepper perform better in an actual task.

References

[1] Thomas Wisspeintner, Tijn Zant, Luca Iocchi, and Stefan Schiffer. Robocup@home: Results in benchmarking domestic service robots. volume 5949, pages 390–401, 06 2009.

[2] Tong Zhang, Chengju Liu, and Qijun Chen. Rebalance control for humanoid walking based on online foot position compensation. pages 4605–4610, 09 2017.

[3] Chengju Liu, Li Xia, Changzhu Zhang, and Qijun Chen. Multi-layered cpg for adaptive walking of quadruped robots. *Journal of Bionic Engineering*, 15:341–355, 03 2018.

[4] Peng Yun, Lei Tai, Yuan Wang, Chengju Liu, and Ming Liu. Focal loss in 3d object detection. *IEEE Robotics and Automation Letters*, PP:1–1, 01 2019.

[5] Changzhu Zhang, Hak Keung Lam, Jianbin Qiu, Chengju Liu, and Qijun Chen. A new design of membership-function-dependent controller for t-s fuzzy systems under imperfect premise matching. *IEEE Transactions on Fuzzy Systems*, PP(99):1–1, 2018.

[6] M. Labbé and F. Michaud. Appearance-based loop closure detection for online large-scale and long-term operation. *IEEE Transactions on Robotics*, 29(3):734–745, June 2013.

[7] Mathieu Labbé and François Michaud. Rtab-map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation: LabbÉ and michaud. *Journal of Field Robotics*, 36, 10 2018.

[8] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *ArXiv*, abs/1804.02767, 2018.

[9] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

7 Team Information

Team Name

TJArk@Home

Contact Information

He Zongtao
Robot and Artificial Intelligence Lab (RAIL)
Tongji University
Caoan Road 4800, Jiading, Shanghai, China
1930719@tongji.edu.cn

Website

https://tjark.cn/file/Pepper/TJArk@Home/

Team Members

He Zongtao, Xu Weihan, Zhou Xun, Liu Zhihao, Deng Xiuqi,
Wang Liuyi, Wang Naijia, Du Jiayuan, Lu Liwen

Description of Hardware

Pepper by Softbank Robotics
Wireness Router
Server computer connected by WiFi

Description of Software

Operation System: Ubuntu 18.04/16.04 LTS; NAOqi OS
Web Server: Tornado
Middleware: ROS
SLAM and Navigation: RTAB-Map
Object Recognition: YOLOv3
Face Dtection and Recognition: YOLOv3
Human Pose Estimation: OpenPose
Speech Synthesis: Xunfei Cloud Service
Speech Recognition: Xunfei Cloud Service