Tinker@Home
2020 Team Description Paper

Jianghao Huo, Xinyu Han, Haocheng Ma, Cuijie Xu, Robin Ananda,
Tengfei Zhang, Xuelin Sun, Zijian Zhang,Dik Hin Leung , Yanjun Jiang,
Shanglin Yang,Ziwei Ning
Tinker(Group), Tsinghua University, China
Homepage: http://tinker2020.furoc.club
Correspondence: robocup-tinker@googlegroups.com

Abstract. In this paper, we describe the joint effort of the Team of Tinker in the past year. RoboCup@HOME consists of a settled set of benchmarking tests that cover multiple skills needed by domestic service robots. We present the hardware chosen, the approaches used and the system established to accomplish the tasks assigned in the paper, and also improvements achieved since RoboCup@HOME 2019. It includes a fromwork for behavior modeling and communication employed between robot and humans, as well as the policy decisions made within the robot itself. We describe our main contributions in arranging the logical model, integrated with various open source algorithm.

Keywords: RoboCup@Home · RoboCup · Domestic service robotics · Tinker@Home

1 Introduction

The RoboCup@Home competition [1] aims at bringing robotic platforms to use in realistic domestic environments. In contrast to other leagues like soccer – which predefine and standardize the playing field – in Robocup@HOME, robots need to deal with different apartment layouts, continuously changing surroundings, unknown sites, unstructured public spaces, and interacting or cooperating with humans who are only very briefly – or not at all, in many situations – instructed about how to interact with the robot. The environment the competition provides varies from year to year. And thus it is typically required that the robots must be capable for multiple tasks, including navigation, human recognition and tracking, speech understanding and simple dialogues, object recognition and manipulation. How could a robot manage all these requirements,

and in the meantime, deals with any possible exception or interference during the tasks is the main problem that teams participating in RoboCup@HOME have to deal with.

Future Robotics Club, or FuRoC team in short, develops a student club that is made up of undergraduates from Tsinghua University, focusing on domestic robots, artificial intelligence tech and the related fields. Robocup@HOME 2020 will be our sixth participation in the @home League of World RoboCup. Our team has come through great reform at the technical research level since last year, and thus the upgraded Tinker is somehow new to the stage.

2 RoboCup@HOME

The RoboCup@Home league [1] aims at bringing robots into domestic environments and support people in daily scenarios.

The leagues are divided into three sub-leagues. Two of them are Standard Platform Leagues for which all competitors use the same robot, and the rest is Open Platform League that grants complete freedom to all competitors. The official leagues and their names are:

- the RoboCup@Home Domestic Standard Platform League (DSPL),
- the RoboCup@HomeSocial Standard Platform League (SSPL), and
- the RoboCup@Home Open Platform League (OPL)

Each league points out to a different aspect of service robotics, reason for which they target specific abilities.

The robots used in the first two Standard Platform Leagues competition are fixed. They aims at benchmarking teams on their integrated software on a common hardware platform, and the standard platform provided to them is mature ones provided by commercial enterprises. Given the target population and anticipate results, the tasks are also nichetargeting.

On the contrary, teams competing in the Open Platform League build the robot themselves. They are provided with the possibility to customize their robots and also the possibility to be portable within the limitation defined by the rulebook.

The competition consists of 2 Stages and the Finals. Each stage consists of a series of Tests that are being held in a daily life environment. The best teams from Stage I advance to Stage II which consists of more difficult tests. The competition ends with the Finals where only the two highest ranked teams of each league compete to select the winner.

In Stage I, robots are tested for their basic functionality, such as finding, object detecting, speech recognition and navigating. In Stage II, they are challenged for their capability of integrating the

functionality in Stage I to deal with more complex environment and tasks. In Final, the top - two teams will display a demonstration that arouses interest in the aspect of future research or application.

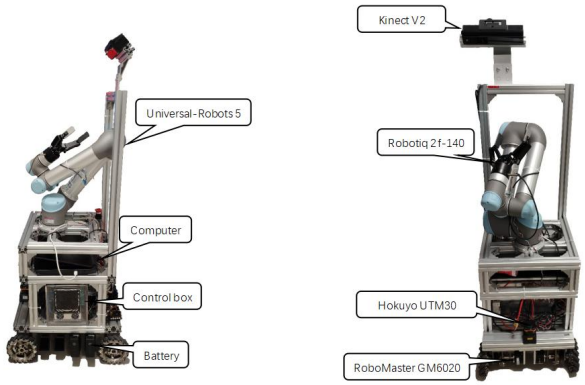

Fig. 1. The robot Tinker in the front view (left) and Tinker in the side view (right). The overall height of Tinker is ≈ 140 cm. The overall weight of Tinker is ≈ 70 kg.

3 Layer

Tinker is designed to be an autonomous humanoid robot mainly for domestic service.

It is equipped with a agile mobile chassis using 4 mecanum wheels, a UR5 robotic arm ,a powerful gripper (Robotiq-2F140), and various types of sensor. Depth cameras (Kinect v2) are used for imaging and recognizing environment, objects and people. Furthermore, an additional camera (realsense d435i) is assembled to the gripper, which assists the head camera while grasping. A single-line laser (UTM-30LX) is used in observing the environment and avoiding obstacles.

From hardware interference to artificial intelligence decision making, Tinker has to deal with challenges at different levels. Thus, a multi-level, distributed structure based on the ROS platform is employed to meet such need. The hardware layer contains an embedded board driving motors and preprocessing odometry data. The hardware-communication layer is responsible for the overall control of the motors and the collection of data from the sensors. The output of the hardware layer is the ROS-compatible-sensor images, including camera images, point clouds and multiple other topics. The logic layer is in charge of providing basic functions of Tinker such as manipulation, navigation, person tracking, object recognition, speech understanding and synthesis etc. The decision layer serves as the collector of topics published by the logic layer and decision maker for the next integrated action to accomplish the task.

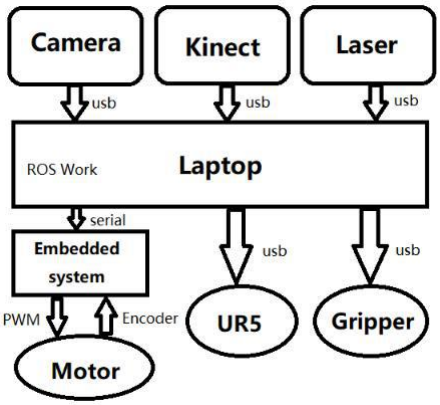

Fig. 2: The theoretical structure of the hardware-communication layer

The layers

·The power management we use are:

1. Dji battery with DCDC transformer for the other equipments.

·The mechanism we use are:

1. DJI GM6020 motors for driving the chassis and the platform.
2. Universal-Robots UR5 robot arm for accessing objects.
3. Robotiq-2f140 mechanical gripper.

·The Sensors we use are:

1. Hokuyo UTM30 laser scanner for navigation
2. Kinect v2 depth camera for navigation and object detection
3. Realsense D435I camera for object recognition and detection
4. Encoder on motors for motor controlling

The hardware-communication layer

The hardware-communication layer must be highly scalable to quickly install and uninstall different executors of the sensors. All control commands of the robot are sent to ROS nodes that are running on the laptop. The laptop also gathers the data collected by the sensors and give orders to the mechanical sections.

The logic layer

The most important functions of Tinker are implemented in this layer.

The main components in this layer include:

1. Navigation: Mapping, localization, route-planning and collision avoidance
2. Vision: Human recognition, object recognition and their tracking.

3. Speech: Speech recognition and synthesis
4. Manipulation: Robot arm planing with feedback from vision and laser

The decision layer

Task planning is done in decision layer. The modules in the decision layer are run as state machines. They integrate the data uploaded from the lower layers to decide which state they are in, and then give different orders or make different responses. Each module deal with just one single task, and share the same information from the lower layers.

4 Hardware

Chassis

Owing to the Mecanum wheels, Tinker is capable of moving in all directions without having to turn around. For safety reason, the speed is limited to no more than 0.25m/s. Compared with the locomotion system in the past, the most important difference is the actuators. The new actuators make far less noise than the former ones because they don't have gearbox inside. Thus Tinker can operate more quietly, which makes it more user-friendly. The chassis consists of 4 separated Mecanum wheel systems, each of which is made up of a Mecanum wheel attached with DJI M3058P19 motor. The laptop sends order to the embedded board to control the chassis as they move along trajectory planed by the ROS Manipulation logic layer. The chassis has a size of 800mm × 500mm × 200mm.

The Arm and Hand

The arm and hand is the most major part of the robot. We choose Universal Robotic 5 (UR5) as its arm for its length and power meets the need to hold most of the objects at home, and Robotiq-2f140 as its hand for it gives a reliable grasping. A Realsense attached at the end of arm is used to object relocalization.

Head Camera & Hand Camera

We choose Kinectv2 as the head camera, providing the central processor with point source and depth images to navigate as well as detect targets, and Realsense D435i as the hand camera, assisting in object detection and recognition.

5 Approach

Human Tracking

For human tracking and following, we implemented the TLD (Track Learning Detection) algorithm [2]. TLD was originally proposed by Zdenek Kalal in 2010. Currently it has developed to be one of the

frontier real time tracking algorithms. Apart from combining the traditional tracking and detecting algorithms, it is more robust when taking into consideration the distortion and partial occlusions. The algorithm consists of three modules as its name indicated. The Tracking Module estimates the moving direction of the object according to the difference between two adjacent frames. The Detection Module detect the object in each frame independently. The Learning Module integrates the results of the Tracking Module and Detection Module to correct the detection errors and update the features of the target object. The algorithm is applied to human tracking and following tasks. Before the robot starts tracking, the human partner that is to be followed will be asked to stand in front of Tinker so that it can record his/her features. Once the record is done, the robot will track and keep up with him. Tinker also uses the depth information to keep itself a safe distance away from the instructor. Moreover, Kinect camera is used to analyze human skeleton, make simple recognition and understand of body language.

Face Recognition

In order to support human-robot interaction, the robot is required to recognize different masters or guests in domestic service. We established a face recognition system with two steps: enrollment and recognition. During the enrollment section, a person will be asked to stand in front of the RGB camera. The face detector based on Haar feature from OpenCV is applied and the detected feature will be stored. For a single being, the system stores 3-5 pictures. We implemented the face recognition algorithm based on sparse representation. A redundant dictionary is trained offline with a set of pre-imported faces. The algorithm seeks the most sparse representation coefficient by solving a L1 optimization problem. The residual errors for different classes (persons) will tell who is the unknown person: if the residual error for a specific class, for example, person A, is smaller than a specified threshold, and the errors for other classes are larger than another specified threshold, the newcoming person is identified as person A. More details of this face recognition pipeline can be found in [4].

Object recognition

Tinker uses a two-phase approach to recognize objects and precisely manipulate them. In the first phase, a point cloud is built according to the features collected by the Kinect depth camera, and we use Fast Plane Extraction in Organized Point Clouds inside. In simple terms, it is to extract the object from the two-dimensional picture, use the ransac and least square method to fit its shape parameters, and use the known three-dimensional spatial position information to reproject it into the three-dimensional space. The real-time plane extraction in 3-dimension point clouds is crucial to many robotics applications. We present an innovative algorithm to reliably detect multiple planes in organized point clouds obtained

from devices - such as Kinect sensors, in real time. By uniformly dividing such a point cloud into non-overlapping groups of points in the image space, we are able to construct a graph in which the nodes and edges represent a group of points and their neighborhood respectively. We then perform an hierarchical clustering on this graph to systematically merge nodes that belong to the same plane until the squared error of the plane fitting mean exceeds a threshold. Finally we refine the extracted planes using pixel-wise region growing. Our experiments demonstrate that the proposed algorithm can reliably detect all major planes in the scene at a frame rate of more than 15Hz (for point clouds generated by 640*480 depth images), which is much faster than many other algorithms we know.
For object classification, another image processing method is implemented. We use fast YOLO [5] for general object type detection, which is a precise while light-weight neural network, designed for general object detection and classification. Then we implemented the Word Bag Model [6] to pair the object image with those collected in the data base.

SLAM

SLAM is an important algorithms that enables a robot to navigate and explore in an unknown environment [7]. The Mapping task requires the robot to record, integrate and update the information it collected about the surroundings while the Localization task requires the robot to identify its own location, refering to the estimated environment. Using a laser range finders (LRFs), we adopted the SLAM package to produce 2D occupancy grid map of the space. The raw data from LRFs are collected as the input stream. Features, or landmarks, are also extracted from the environment. When Tinker moves around, these features are used to estimate its location. It is called the Laser-Scan-Matcher process. However, the estimation of this process is imprecise and the error accumulates. The GMapping process is adopted [8], assisting by the EKF (Extended Kalman Filter) to correct the estimated result and draw the map [9].

Navigation

Navigation is one of the basic capability that a domestic robot must acquire. Based on the generated map, the robot needs to plan the route from its current position to the target one. Considering both distance and collision avoidance, the A* algorithm is adopted to find the route. Moreover, the robot must be able to handle unexpected obstacles when moving around. Thus, the navigation package is applied and modified for Tinker so that parameters in the move_base package are tuned and the navigation task can be achieved functionally. However, the behavior and speed is far from satisfactory. We now extend a local package which subscribes the global plan from the origin and linearizes the curve. In this way, the whole process becomes much more fluent. To avoid small objects

and non-cylinder-like objects like chairs and cups on the floor, the robot is equipped with depth cameras - including a kinect2 and - to build another local obstacle layer. Since the pointcloud tend to be noisy, we introduce a filter to this obstacle layer to achieve more stable navigation performance, and a social layer is added to classify Bayesian data. Once a person enters the sight of the camera, he will be identified something alive and tagged. Even if he leaves the sight of the camera, he will be marked in the clustering model formed by radar to provide better effect for obstacle avoidance.

Speech Recognition & Understanding

For 2020 competition, we implement speech interaction system based on Google TTS(Text-To-Speech) and STT(Speech-To-Text) Cloud API [10]. To overcome the delay of communication between the robot and cloud platform, we import a stream based on audio transform method for speech recognition. Moreover, Tinker caches a huge amount of audio response template locally to speed up the robot reaction.

Apart from the Speech-to-text layer, the dialogue system also contains a simple keyword parser, which takes keywords in certain patterns to operate task switches. Once the software recognizes a sequence with one of the predefined patterns, the robot will interpret one’s intention and makes responses.

6 Summary & Prospect

In this paper we introduced our team and the robot, gave a brief outlook of RoboCup@Home, described the layer of Tinker, and its hardware and software system. A major focus was set on the description of the algorithm approaches. We proposed an novel approach for detecting multiple planes in organized point clouds which we proved to be more efficient. We now focus on two topics: computer-human interaction through gestures and facial expressions and the visual SLAM loop closure detection based on neural network. For the former one, the majority of the researches and applications now put emphasis on how the robot can better understand human, while actually, communication is something bi-directional. Thus, how human can understand what the robot want to “express” is also crucial. Moreover, apart from words, gesture is a very important method to exchange ideas. For the latter one, compared with the traditional Bag of Words algorithm, the design of the neural network-based relocation algorithm is greatly simplified. The idea is more natural, and it is hoped to obtain better performance than the traditional relocation algorithm.

7 Reference

1. Mauricio Mata" moros, Alexander Moriarty, Justin Hart, and Hiroyuki Okada. “Robocup@Home 2020: Rule and regulations,” https://athome.robocup.org/wp-content/uploads/2020_rulebook.pdf, 2020.
2. Z. Kalal, K. Mikolajczyk, and J. Matas, “Tracking-learning-detection,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 7, pp. 1409–1422, 2012.
3. J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, “Robust face recognition via sparse representation,” Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 31, no. 2, pp. 210–227, 2009.
4. F. Xia, L. Tyoan, Z. Yang, I. Uzoije, G. Zhang, and P. A. Vela, “Human-aware mobile robot exploration and motion planner,” in SoutheastCon 2015. IEEE, 2015, pp. 1–4.
5. J. Redmon and A. Farhadi, “Yolo9000: Better, faster, stronger,” 2016.
6. G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, “Visual categorization with bags of key-points,” in Workshop on statistical learning in computer vision, ECCV, vol. 1, no. 1-22. Prague, 2004, pp. 1–2.
7. G. Grisetti, C. Stachniss, and W. Burgard, “Improved techniques for grid mapping with rao-blackwellized particle filters,” Robotics, IEEE Transactions on, vol. 23, no. 1, pp. 34–46, 2007.
8. Giorgio Grisetti, Cyrill Stachniss, and Wolfram Burgard: Improving Grid-based SLAM with Rao-Blackwellized Particle Filters by Adaptive Proposals and Selective Resampling, In Proc. of the IEEE International Conference on Robotics and Automation (ICRA), 2005.
9. Giorgio Grisetti, Cyrill Stachniss, and Wolfram Burgard: Improved Techniques for Grid Mapping with Rao-Blackwellized Particle

Filters, IEEE Transactions on Robotics, Volume 23, pages 34-46, 2007.

10. P. Lamere, P. Kwok, E. Gouvea, B. Raj, R. Singh, W. Walker, M. Warmuth, and P. Wolf, “The cmu sphinx-4 speech recognition system,” in IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP 2003), Hong Kong, vol. 1. Citeseer, 2003, pp. 2–5.

8 Team Repository

Our team repository can be found at https://github.com/tinkerfuroc.
The repository may be of help to other teams by providing:

1. Implementation of all the algorithms and needed parameters described in the paper
2. Robot setup scripts and tools
3. Code for RoboCup@Home tasks

9 3rd Party Dependencies

1. ROS
2. tensorflow
3. Face++
4. Google Cloud API
5. OpenCV
6. PCL

10 Acknowledgement

The authors of this paper would like to thank previous team members of Tinker@Home2014, Tinker@Home2015, Tinker@Home2017, Tinker@Home2019, for their help and support through out building the robot and writing this manuscript. Particular thanks to old members Jingsong Peng, Jiacheng Guo and Yilin Zhu. The authors would like to thank teachers Yanxiong E, Xiaoliang Zhu in Youth League Committee of Tsinghua University.

9 Contact Info

Address: Tsinghua University,Hai Dian, Beijing
Email: jackeyhuo15@gmail.com