Tinker@Home 2023 Team Description Paper

Xinyao Qin, Haoyue Liu, Robin Ananda, Tengfei Zhang, Zhi Wang,
Yunfei Li, Qihan Guo, Zhiting Zhou, Qingwei Ben, Tingxuan Leng,
Guozhi Li, Fangzhou Li, Hongyi Cai, Dekun David Kong,
Yaxi Jiang, Enfei Yu, Shanglin Yang

February 7, 2023

Abstract. In this paper, we provide our robot system design details, including software and hardware, for RoboCup@home competition. We also show our improvements since last year. Our system design includes a framework of behavior modeling and human robot interface. Our main contributions are a novel logic model, and a framework to integrate various open source algorithms.

1 Introduction

The RoboCup@home competition aims at providing an ideal environment for robot learning especially self-supervised and continual learning. In contrast to the mainstream supervised learning which based on large amount pre-annotated data, in RoboCup@home, robots is required to learn continuously in perception, cognition, manipulation, and interaction with users by unsupervised learning.

Future Robotics Club, or FuRoC team in short, is a student club of undergraduates from Tsinghua University, focusing on domestic robots, artificial intelligence and related fields. We have participated in Robocup@Home many times in the history and achieved pretty good results, Rank 5 in 2016 and Rank 7 in 2019. Robocup@Home2023 will be our 8th participation in the @Home League of World RoboCup. Our team made remarkable technical improvements since last year, and thus upgraded Tinker to a new stage.

2 Layer

Tinker is designed to be an autonomous 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 (Azure Kinect DK) are used for imaging and recognizing environment, objects and people. Furthermore, a single-line laser (UTM-30LX) is used in observing the environment and avoiding obstacles with the depth cloud from camera. A iFlytek microphone is also attached to the robot for human-robot interaction.

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. There are four layers in total on Tinker, namely the hardware layer, the hardware-communication layer, the logic layer and the decision layer. 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-communication 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.

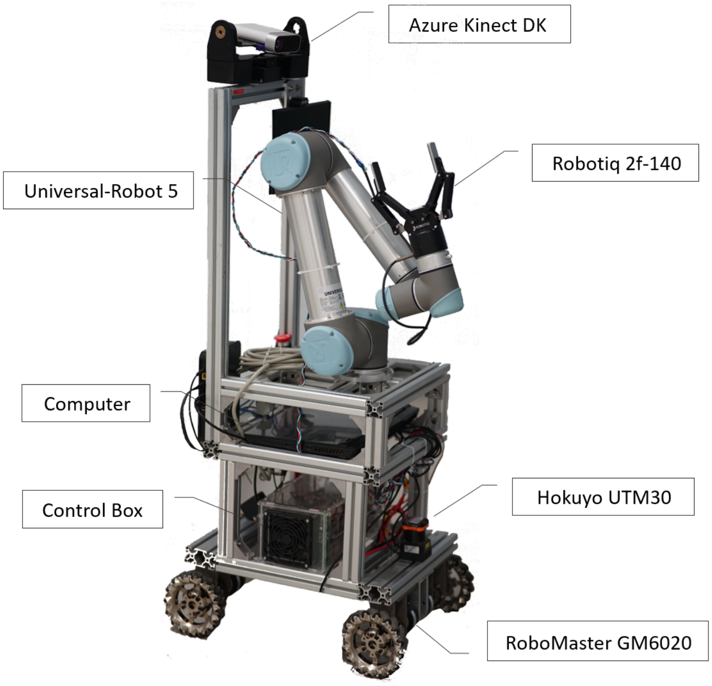

Fig. 1. The robot Tinker in the side view (right). The overall height of Tinker is ≈ 140 cm. The overall weight of Tinker is ≈ 70 kg.

2.1 The hardware layer

Power management modules:

- Dji battery with DCDC transformer for the other equipments.

Actuators:

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

Sensors:

- Hokuyo UTM30 laser scanner for navigation.
- Azure Kinect DK depth camera for navigation and object detection.
- Realsense D435I camera for object recognition and detection.
- Encoder on motors for motor controlling.

2.2 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.

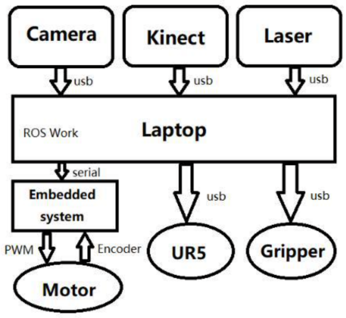

Fig. 2. Structure of the hardware-communication layer

2.3 The logic layer

The most important functions of Tinker are implemented in this layer. The main components in this layer include:

- Navigation: Mapping, localization, route-planning and collision avoidance.
- Vision: Human recognition, object recognition and their tracking.
- Speech: Speech recognition and synthesis.
- Manipulation: Trajectory planning, Motion control and Grasping with force feedback.

2.4 The decision making 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.

3 Hardware

3.1 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. 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.

3.2 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.

3.3 Head & Hand Camera

We choose Azure Kinect DK 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.

4 Approach

4.1 Vision

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 [1] 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 [2] to pair the object image with those collected in the data base.

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 [3].

Human Tracking For human tracking and following, we implemented the TLD (Track Learning Detection) algorithm [4]. 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.

4.2 Manipulation

In order to complete the tasks of delivering things, Tinker needs to finish two related subtasks, montion planning and grasping. With a 6-DOF UR5 arm, Tinker can reach amlost everywhere fexibly in the three-dimensional space. Tinker carries out its montion planning mainly with the help of MoveIt, using the default OMPL algorithm and Rucking algorithm for its path planning and trajectory generation respectively. To avoid the potential collision with other parts, we restrict the work space of UR5 with yaml files. Tinker also converts the point clouds from the camera into three-dimensional information to avoid these obstacles in path planning. In the process of grasping, we pre-set the angle of the claw closure and the maximum force to ensure that the grasp force will not damage the objects. The built-in current feedback in Robotiq also plays a role in this process.

4.3 Navigation

SLAM SLAM is an important algorithm that enables a robot to navigate and explore in an unknown environment [5]. 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. We adopted the loop-closure and optimization process in Cartographer [6] to reduce the accumulated error and draw the map. Moreover, considering the obstacles may not be able to be fully detected by the single-line LRF, which can only conduct 2D SLAM, We re-project the point cloud detected by the depth camera onto the map from 2D SLAM in real time to facilitate the mapping.

Route Planning 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 kinect camera and - to build another local obstacle layer. Since the point cloud 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.

4.4 Speech Recognition & Understanding

For 2023 competition, we implement speech interaction system based on Iflytek Six microphone array and its online speech recognition and synthesis module. 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.

5 Conclusion

In this paper we introduced our team and the robot, 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.

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. The authors would like to thank teachers Li Liu and Geng Lu, who gave a lot of guidance and help to this team.

References

1. J. Redmon and A. Farhadi. Yolo9000: Better, faster, stronger. 2016.
2. G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of key-points. *Workshop on statistical learning in computer vision, ECCV*, 1, 2004.
3. J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):210–227, 2009.
4. Z. Kalal, K. Mikolajczyk, and J. Matas. Tracking-learning- detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7):1409–1422, 2012.
5. Giorgio Grisetti, Cyrill Stachniss, and Wolfram Burgard. Improved techniques for grid mapping with rao-blackwellized particle filters, ieee transactions on robotics, volume 23. 2007:34–46.
6. Wolfgang Hess, Damon Kohler, Holger Rapp, and Daniel Andor. Real-time loop closure in 2d lidar slam. *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1271–1278, 2016.

Appendix

A.Team Repository

Our team repository can be found at https://github.com/tinkerfuroc. The repository may be helpful 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 tasks

B.3rd Party Dependencies

- ROS
- Tensorflow
- Face++
- Google Cloud API
- OpenCV
- YOLO
- PCL
- MoveIt