Tinker@Home 2019 Team Description Paper

Jianghao Huo, Xinyu Han, Junda Bi, Haocheng Ma, Renjie Ma, Cuijie Xu, Lubin Ye, Han Zhang, Tengfei Zhang, Zijian Zhang and Yao Jiang

Future Robotics Club(Group), Tsinghua University, China
Homepage: http://tinker.furoc.net
Correspondence: robocup-tinker@googlegroups.com

Abstract. This paper describes an intelligent robot named Tinker of Tsinghua University, China, including the mechanical system, hardware system and software system. Tinker is designed to be an autonomous robot in domestic environment, capable of navigating and manipulating in complicated environment with friendly interaction interface, aiming to accomplish the tasks of @Home League of World RoboCup 2019. This paper gives the hardware and software design we have proposed and implemented.

1 Introduction

Automatic home service is a research hotspot these years, which calls for a more intelligent and reliable robot. With the best interest in robot, Tinker is created by FuRoC team (Future Robotics Club), a student club from Tsinghua University focusing on robotics, AI and related fields. Robocup@HOME 2019 is our forth participation in the @home League of World RoboCup. Tinker is designed to be an autonomous humanoid robot mainly for home service. The capability such as automatic navigation, environment perception, interaction with human, vision recognizing and objects carrying, etc, are necessary. Tinker is equipped with a agile mobile chassis using 4 mecanum wheels, a UR5 robotic arm ,a powerful mechanical gripper (Robotiq-G85), and various types of sensor. Depth cameras (Kinect v2 and realsense d435i) are used for imaging and recognizing environment, objects and people. 2 laser range-finders and 1 3D Lidar are used for environment perception and obstacle avoidance. To confirm the objects grabbing, the gripper is equipped with a Laser transmitter and receiver sensor.

2 Overview of the robot

2.1 Overview of the architecture

Tinker system has to deal with challenges at different levels from hardware interface to artificial intelligence. Thus, a multi-level, distributed architecture based on ROS is employed to meet such requirement. The hardware layer contains a embedded board driving motors and preprocessing odometry data. The hardware-communication layer is responsible for controlling the motors and acquiring information from the sensors. The output of the hardware layer is ROS-compatible sensor images including camera image, point cloud and multiple other topics. The logic layer is responsible for providing basic functions of a robot such as manipulation, navigation, human tracking, object recognition, speech recognition and synthesis etc. The decision layer listens to topics published by the logic layer and makes decision for the next high-level action to accomplish the task.

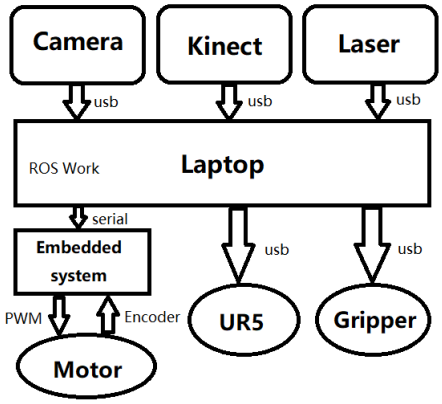


Fig. 1: The architecture of the hardware-communication layer

The hardware layer

The power management we use are:

1. Calf electric vehicle battery for center processor and chassis.
2. Dji battery with step-down or up transformer for the other equipments.

The mechanism we use are:

1. DJI M3058P19 motors for driving the chassis and the platform.
2. Universal-Robots UR5 robot arm for accessing objects.
3. Robotiq-G85 mechanical gripper.

Sensors we use are:

1. Hokuyo URG04LX laser scanner for navigation
2. Hokuyo UTM30 laser scanner for navigation
3. VEILODYNE VLP16 multi-laser scanner for navigation and hunamn tracking
4. Kinect v2 depth camera for navigation and object detection
5. Realsense D435I camera for object recognition and detection
6. Xsense MTi-10 nine-axis IMU
7. Encoder on motors for motor controlling
8. Laser transmitter and receiver sensor for confirming the grasping

The hardware-communication layer

The hardware communication layer must be highly scalable to quickly install and remove different sensors and executors. All control commands of the robot are sent to the ROS nodes running on the laptop. Laptop also collect the information collected by the various sensors and control mechanical operations.

The logic layer

Most important robot functions 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. For different tasks, modules in decision layer run as state machine. They integrate different information from the low layer to judge the state they are in and then give different orders or make different responses. Each module deal with a single task, sharing the common information from the lower layers.

3 Mechanical Design

To complete most of home serving tasks, our robot consists of three major parts: chassis based on Mecanum wheel, UR5 arm including Robotiq-G85 hand. Tinker is about 140cm in height.

Chassis Tinker can move in any direction easily owing to the Mecanum chassis. The chassis consists of 4 separated Mecanum wheel systems, each of which consists of a Mecanum wheel attached with DJI M3058P19 motor. The PC sends control message to the embedded board to command the chassis to move along trajectorys planed by ROS Manipulation logic layer. The chassis has a size of 800mm $\times$ 500mm $\times$ 200mm.

Robot arm and hand The robot arm and hand are the most major part of the mobile robot, used to grasp objects.UR5 is long and powerful enough to hold most of the objects at home, and Robotiq-G85 gives a reliable grasping.A laser transmitter and receiver sensor is used to confirm the grasping.

4 Software Engineering

4.1 Computer Vision

Computer vision is indispensable for tinker to accomplish multiple tasks including person recognition, object manipulation and environment modeling.

Human Tracking For human tracking and following, we implemented the TLD (Track-Learning Detection) algorithm[1]. TLD was proposed by Zdenek Kalal and is currently the state-of-art real time tracking algorithm. It combine the traditional tracking and detection algorithm so that it is more robust in consideration of distortion and partial occlusions. TLD algorithm consists of three modules as its name indicated. Tracking module estimate moving direction of the object according to the difference between two adjacent frames. Detection module detect the object in each frame independently. Learning module integrate the results of the tracking module and detection module to correct the detection errors and update the features of the target object. We applied the TLD algorithm to human tracking and following tasks. Before the robot starts following, the human partner to be followed will

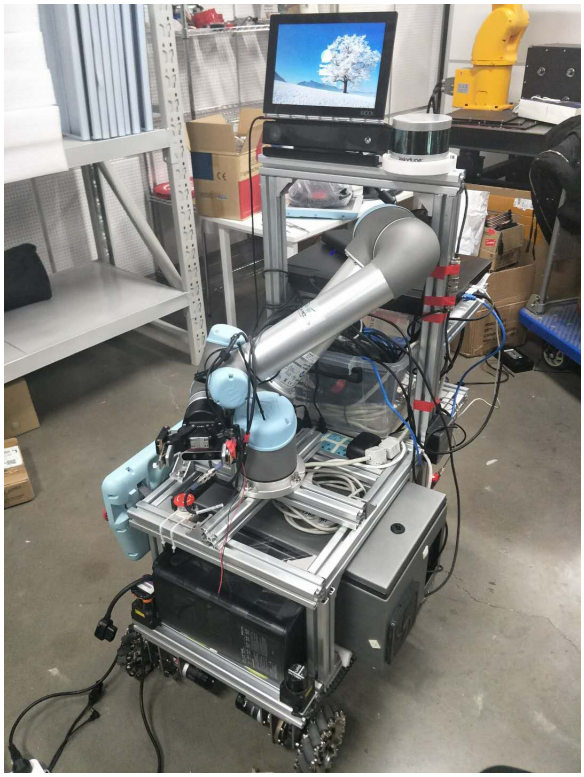


Fig. 2: Picture of tinker robot

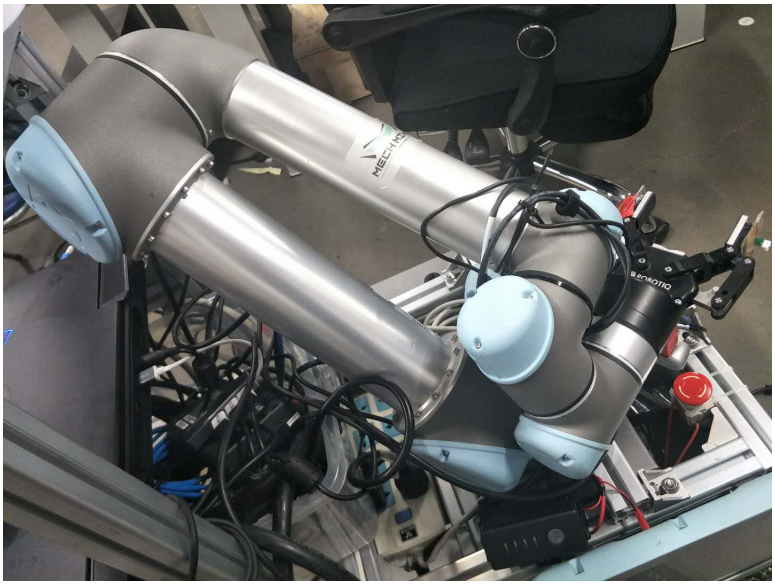


Fig. 3: Robotic Arm and Hand

be asked to stand in front of Kinect and the robot will record his/her features. When the instructor starts moving around, the robot will track and keep up with him. The robot also uses the depth information to keep away from the instructor at a safe distance. Moverover, we use Kinect camera to analyze human skeleton and make simple analysis and judgment of body language.

Face Recognition For human-robot interaction, a robot is required to recognize different masters or guests in home service. We developed a face recognition system with two process: enrollment and recognition. During the enrollment process, a man is asked to stand in front of the RGB camera. A face detector based on haar feature from OpenCV is applied and the detected face will be stored. For a single person, the system stores 3-5 pictures. We used face++ supported by Questyle Audio for face detection and implemented the face recognition algorithm based on sparse representation [2]. A redundant dictionary is trained offline using a set of training faces. The algorithm seeks the most sparse representation coefficient by solving a L1 optimization problem. The residual errors for different classes (persons) 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. Fig.4 shows an example of the face recognition result. More details of this face recognition pipeline can be found in [3].

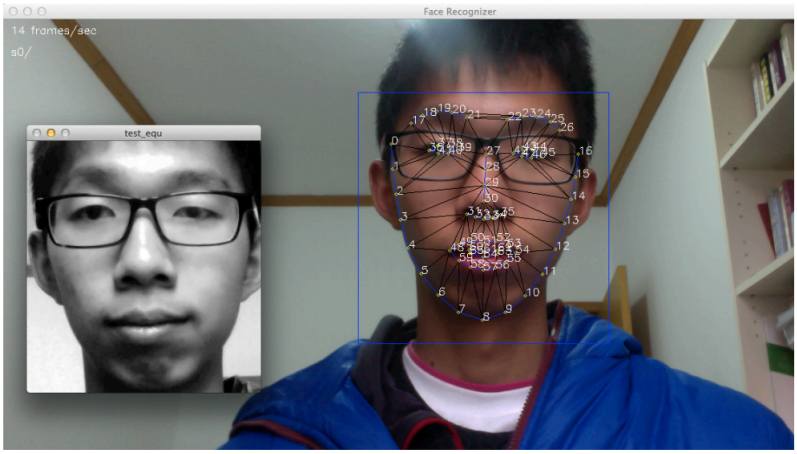


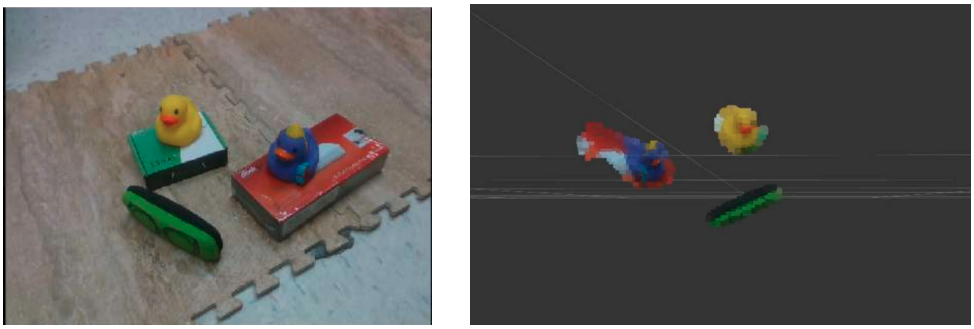
Fig. 4: face recognition

Object recoginition

Tinker uses a two-phase approach to recognize objects and precisely manipulate them. In the first phase, a point cloud is built from the Kinect depth camera to collect the features, and we use Fast Plane Extraction in Organized Point Clouds Using Agglomerative Hierarchical Clustering inside. Real-time plane extraction in 3D point clouds is crucial to many robotics applications. We present a novel algorithm for reliably detecting multiple planes

in real time in organized point clouds obtained from devices such as Kinect sensors. By uniformly dividing such a point cloud into non- overlapping groups of points in the image space, we first construct a graph whose node and edge represent a group of points and their neighborhood respectively. We then perform an agglomerative hierarchical clustering on this graph to sys- tematically merge nodes belonging to the same plane until the plane fitting mean squared error 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 35Hz for 640*480 point clouds, which to the best of our knowledge is much faster than state-of-the-art algorithms.

For object classification, another image processing pipeline is implemented. We used fast YOLO [4] for general object type detection. fast YOLO is a light-weight while precise nerual network for genral object detection and classification. Then we implemented a bag of words model [5] to pair the object image with those collected in the library.



(a) Origin (b) Result

Fig. 5: Objects recoginitions show

4.2 Navigation

Simultaneous Localization and Mapping SLAM is one of the most important algorithms for a mobile autonomous robot, which enables a robot to navigate and explore in an unknown environment [6]. Mapping requires the robot to record, integrate and update the former information it have got about the surroundings while Localization requires the robot to know the location of itself refer to the estimated environment. Using a laser range finders (LRFs), we adopted the SLAM package to estimate the location and its surroundings in the form of 2D occupancy grid map. The raw data from LRFs are collected as the input of the algorithm. Features, or landmarks are then extracted from the environment. When the robot moves around, these features are used to estimate where it moves. It is called Laser-Scan-Matcher process. However, the estimation of this process is imprecise and the error accumulates. The GMapping process is adopted, using an EKF (Extended Kalman Filter) to correct the estimated result. Based on the final map generated, the robot plans its path and explores the unknown environment. We also implemented SLAM using color and depth camera, also called vSLAM [7], so that a 3D map can be obtained, which is more precise

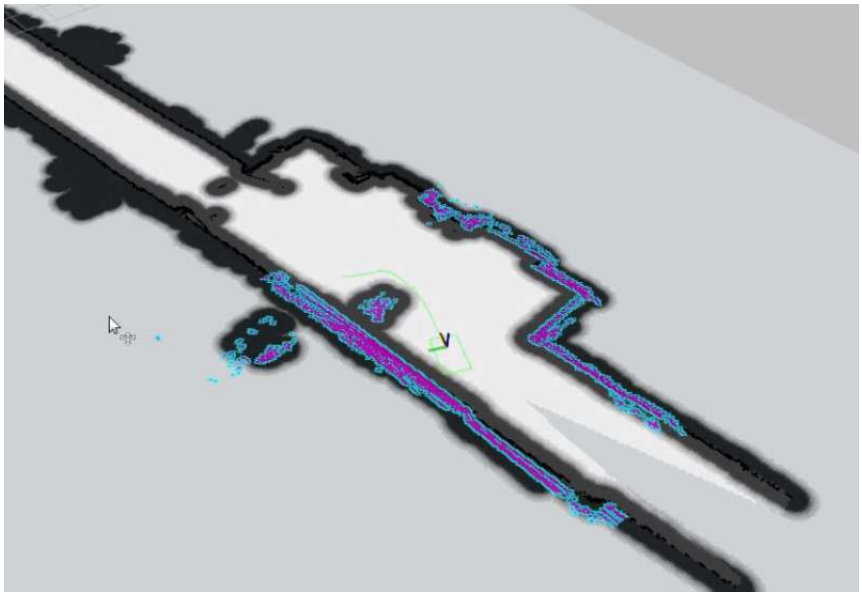


Fig. 6: Navigation show

in complicated environment. Building map using vSLAM is still a experimental feature for tinker and needs further refining.

Navigation Navigation is one of the basic function that a mobile autonomous robot must have. The robot needs to plan the route from its current position to the goal. An A* algorithm is used to find the route considering both distance and collision avoidance. Moreover, the robot must be able to handle unexpected obstacles when moving around. The navigation package is applied and modified for the tinker robot. Parameters in the move_base package are tuned and the navigation task can be achieved functionally but the behavior and speed is far from satisfactory. We extended a local which subscribes the origin global plan and linearizes the curve. In this way, the whole processing could be more fluently. To avoid small objects and non cylinder-like objects like chairs and cups on the floor, we use depth cameras including a kinect2 and a primesense to build another local obstacle layer. Since pointcloud tend to be noisy, we filter this obstacle layer to achieve more stable navigation performance. This year, a new social layer was added to classify Bayesian data. When a person enters the camera line of sight, he will be judged to be living and marked. Even if he leaves the camera line of sight, he will be tagged in the clustering model formed by radar to provide better effect for obstacle avoidance.

4.3 Speech Recognition

For 2019 competition, we implement speech interaction system based on google TTS(Text-To-Speech) and STT(Speech-To-Text) Cloud API. To overcome communication delay from robot and cloud platform, we realize a stream based audio transfrom method for speech recognition. Moreover, Tinker caches a huge amount of audio response template locally to accelerate robot correspondance.

After speech to text layer, dialogue system consists of a simple keyword parser, which takes keywords in certain patterns to operate task switches: When the software recognizes a sequence with one of the predefined patterns, the robot interprets one's intention and makes corresponding responses. Sound source localization method is realized by IFLYTEK develop toolkit.

References

1. 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.
2. 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.
3. 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.
4. J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," 2016.
5. G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Workshop on statistical learning in computer vision, ECCV*, vol. 1, no. 1-22. Prague, 2004, pp. 1–2.
6. 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.
7. S. Se, D. G. Lowe, and J. J. Little, "Vision-based global localization and mapping for mobile robots," *Robotics, IEEE Transactions on*, vol. 21, no. 3, pp. 364–375, 2005.
8. 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.

5 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

6 3rd Party Dependencies

1. ROS
2. ROS Navigation Stack
3. iai_kinect2
4. darknet(tensorflow fast yolo)
5. tensorflow
6. face++

Acknowledgement

The authors of this paper would like to thank previous team members of Tinker@Home 2014, Tinker@Home2015, Tinker@Home2017 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 Wenhao Ding, Laixi Shi, Zhaoyuan Gu ,Gang Li and Shuo Yang, our lead teachers Yu Wang and Yao Jiang, teachers Yanxiong E, Xiaoliang Zhu in Youth League Committee of Tsinghua University and would specially thank Mech-Mind, for their exclusive sponsorship and Questyle Audio for their technology support. At last, Team leader Han Zhang's personal thanks to Linbo Fu: without her help, I may never be able to lead this team.

Contact Info

Address: Tsinghua University,Hai Dian, Beijing
Email: jiangyaonju@126.com