RoboCup UTHM *MNRS*

Herdawatie binti Abdul Kadir, Abu Ubaidah bin Shamsudin, Hazwaj bin Mhd Poad, Johan Lela Andika Bin Johan Budiman, Muhammad Amirul Amri Adam, Muhamad Faizul Akmal Bin Hamdan, Siti Nurlina Binti Noor Azmi, Nurul 'Ain Fatihah Binti Ahmad Nazri, Nur Naimah Binti Zainal

23 May 2021

https://sites.google.com/view/robocuputhm/home

Abstract. This paper describe an intelligent robot that are focusing on indoor environment. that can map the environment using ros gmapping package. Using the recorded map, a navigation algorithm is used to move to a given goal. The navigation algorithm is combination of Advance Monte Carlo for localization, A* path-planning for global planner and Dynamic Window Approach for local planner. Python google speech recognition package is use convert word to text and python google speec synthesis is use to convert text to sound. For recognition, Speed Up Robust Features (SURF) is used to recognize objects. The robot is design as an autonomous robot that are complete with contribution of combining the ***MNRS*** which are ***mapping, navigation, object recognition and speech recognition***. Our research also provide software system development that are implemented in our robot. This project also motivates the undergraduate students level in the field of computer programming and human-robot interaction.

1.0 Introduction

Our robot design use turtlebot Kobuki as the mobile robot platform. Turtlebot Kobuki is Turtlebot2 type robot consists of Yujin Kobuki base, a 2,200mAh battery pack, a Kinect sensor, an Asus 1215N laptop with a dual core processor, fast charger, charging dock, and a hardware mounting kit attaching everything together and adding future sensors. Turtlebot2 was released on Oct 2012 [1].

2.0 Software

2.1 Mapping

gmapping is one of the algorithm under the SLAM method that are widely used in mapping an environment[2]. Mapping our environment using *keyop* command and moved the robot carefully so that the robot does not hit any object or wall. Run *RViz* to see what the robot see in its camera. After that, save the map so that the turtlebot can move autonomously in navigation stack.

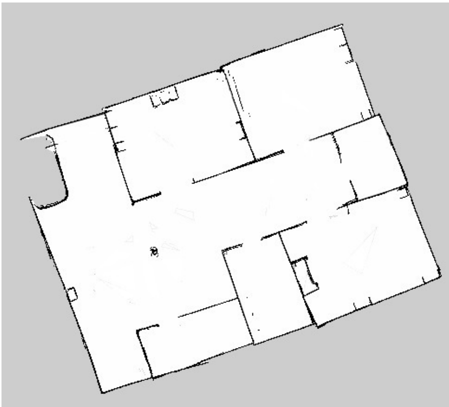


Figure 2.1 shows the mapping after edit

2.2 Navigation

AMCL (Adaptive Monte Carlo Localization) is one of other algorithm under SLAM method in mapping and navigation. AMCL is the expectation of exact location for the robot to move in 2 dimension[3]. **A***is route planning for the robot to avoid hitting with the obstacles from the starting point to the end point[4]. **DWA (Dynamic Window Approach)** a path planning based on robot dynamic motion[5]. The turtlebot navigation is used for robot navigation with map building using *gmapping* and *amcl* while running the navigation stack in ROS. With a complete map we can predefine location point for the robot to move when instructed.

In *RViz*, when starting up with navigation stack, localize the turtlebot using *2D Pose Estimate* to provide the robot its approximate location on the map. Then, it can move autonomously using *2D Nav Goal.*

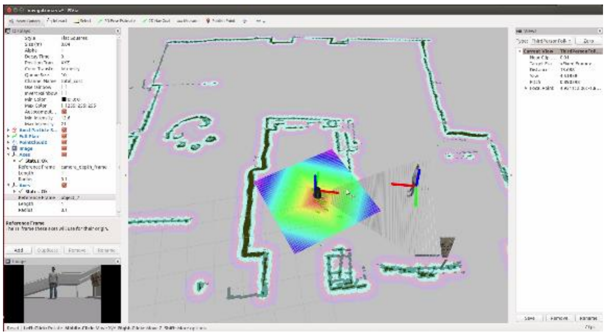


Fig. 2.2. shows the navigation of turtlebot

2.3 Object Recognition

SURF (Speeded-Up Robust Features) is an upgraded version of algorithm keypoint detection and description called SIFT. The complexity of SURF can be determined efficiently with the hand of integral images and Box Filter[6].

Open the gazebo with our custom map as the environment and implemented the objects position through *RViz*. We arranged the item as follows to facilitate recognition: sensor cloudpoint and rbg-> camera on *RViz* was added to defining object in front of camera. We use find- object app to scan people and object then axes was put on people and object based on the number of scanning and axes on turtlebot camera. Finally, turtlebot can detect people around the map when it move around.

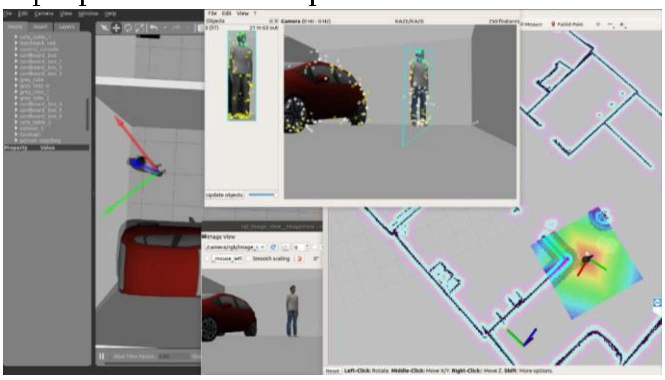
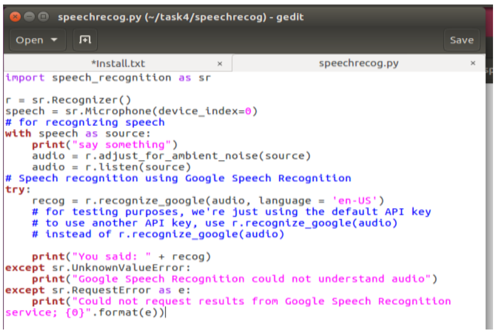


Figure 2.3. show the image in front turtlebot Kaboki's camera

2.4 Speech Recognition

Speech recognition or Automatic Speech Recognition (ASR) is a process to provide an interaction between a human and a machine. Speech recognition file had been installed from attached file inside google drive link named *speech recog.py*

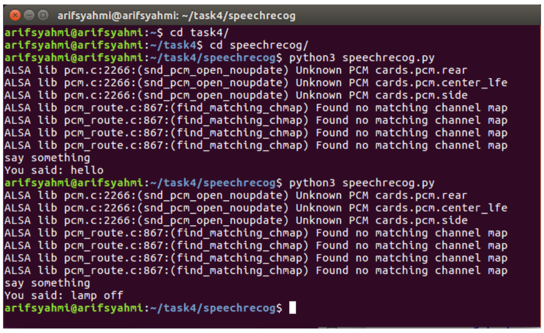
The sound is recorded and saved into speech. By using this speech, ambient noise had been adjusted to get a better sound before we put them into audio. Next, the audio was sent to Google Speech recognition. We started to run the speech by using *python3 speechrecog.py*. At this point, the detected audio will print out as a sentences as shown in Figure 2.4.



```
speechrecog.py (~/task4/speechrecog) - gedit
Open    Save
*Install.txt    speechrecog.py
import speech_recognition as sr

r = sr.Recognizer()
speech = sr.Microphone(device_index=0)
# for recognizing speech
with speech as source:
    print("say something")
    audio = r.adjust_for_ambient_noise(source)
    audio = r.listen(source)
# Speech recognition using Google Speech Recognition
try:
    recog = r.recognize_google(audio, language = 'en-US')
    # for testing purposes, we're just using the default API key
    # to use another API key, use r.recognize_google(audio)
    # instead of r.recognize_google(audio)

    print("You said: " + recog)
except sr.UnknownValueError:
    print("Google Speech Recognition could not understand audio")
except sr.RequestError as e:
    print("Could not request results from Google Speech Recognition
service; {0}".format(e))
```

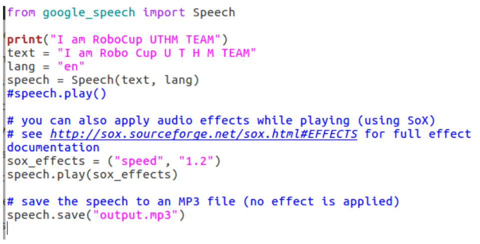


```
arifsyahmi@arifsyahmi: ~/task4/speechrecog
arifsyahmi@arifsyahmi:~$ cd task4/
arifsyahmi@arifsyahmi:~/task4$ cd speechrecog/
arifsyahmi@arifsyahmi:~/task4/speechrecog$ python3 speechrecog.py
ALSA lib pcm.c:2266:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.rear
ALSA lib pcm.c:2266:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.center_lfe
ALSA lib pcm.c:2266:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.side
ALSA lib pcm_route.c:867:(find_matching_chmap) Found no matching channel map
ALSA lib pcm_route.c:867:(find_matching_chmap) Found no matching channel map
ALSA lib pcm_route.c:867:(find_matching_chmap) Found no matching channel map
ALSA lib pcm_route.c:867:(find_matching_chmap) Found no matching channel map
say something
You said: hello
arifsyahmi@arifsyahmi:~/task4/speechrecog$ python3 speechrecog.py
ALSA lib pcm.c:2266:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.rear
ALSA lib pcm.c:2266:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.center_lfe
ALSA lib pcm.c:2266:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.side
ALSA lib pcm_route.c:867:(find_matching_chmap) Found no matching channel map
ALSA lib pcm_route.c:867:(find_matching_chmap) Found no matching channel map
ALSA lib pcm_route.c:867:(find_matching_chmap) Found no matching channel map
ALSA lib pcm_route.c:867:(find_matching_chmap) Found no matching channel map
say something
You said: lamp off
arifsyahmi@arifsyahmi:~/task4/speechrecog$
```

Input Output

Figure 2.4 shows the input and output from audio to sentences

The next stage are converting text to voice. The python3 had been installed from the attached file named text2voice from the same google drive link. From the same file we opened the programmed file and imported *google_speech* into speech. The language used are in English, "en" and the speed adjusted to "1.2". The voice then can be heard clearly. Lastly, we saved the speech to MP3 file named output.mp3.

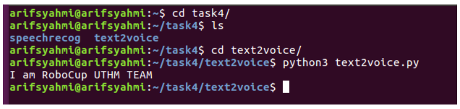


```
from google_speech import Speech

print("I am RoboCup UTHM TEAM")
text = "I am Robo Cup U T H M TEAM"
lang = "en"
speech = Speech(text, lang)
#speech.play()

# you can also apply audio effects while playing (using SoX)
# see http://sox.sourceforge.net/sox.html#EFFECTS for full effect
documentation
sox_effects = ("speed", "1.2")
speech.play(sox_effects)

# save the speech to an MP3 file (no effect is applied)
speech.save("output.mp3")
```



```
arifsyahmi@arifsyahmi:~$ cd task4/
arifsyahmi@arifsyahmi:~/task4$ ls
speechrecog  text2voice
arifsyahmi@arifsyahmi:~/task4$ cd text2voice/
arifsyahmi@arifsyahmi:~/task4/text2voice$ python3 text2voice.py
I am RoboCup UTHM TEAM
arifsyahmi@arifsyahmi:~/task4/text2voice$
```

Input Output

Figure 2.4.2 shows the input and output from text to voice

3.0 Conclusion

In this paper, we presented on how we are integrate ***MNRS*** software and simulation to be used for RoboCup@Home Education 2021 competition.

4.0 Reference

1) Evan Ackerman (2012), "What is Turtlebot?",IEEE Spectrum, https://www.turtlebot.com/about/
2) Norzam, W. A. S., Hawari, H. F., & Kamarudin, K. (2019), "Analysis of Mobile Robot Indoor Mapping using GMapping Based SLAM with Different Parameter", *IOP Conference Series: Materials Science and Engineering*, *705*, 012037. https://doi.org/10.1088/1757-899x/705/1/012037
3) Brian P.Gerkey (2020), "amcl", ROS.org, http://wiki.ros.org/amcl
4) *Path planning*. (2021). Husarion Docs. https://husarion.com/tutorials/ros-tutorials/7-path-planning/
5) D. Fox, W. Burgard and S. Thrun, "The dynamic window approach to collision avoidance", in IEEE Robotics & Automation Magazine, vol. 4, no. 1, pp. 23-33, March 1997, doi: 10.1109/100.580977. https://www.trossenrobotics.com/interbotix-turtlebot-2i-mobile-ros-platform.aspx
6) Alexander Mordvintsev & Abid K. Revision 43532856. (2013), " Introduction to SURF(Speeded-Up Robust Features)", OpenCV-Python Tutorials, https://github.com/abidrahmank/OpenCV2-Python-Tutorials/blob/master/source/py_tutorials/py_feature2d/py_surf_intro/py_surf_intro.rst
7) *Interbotix Turtlebot 2i Mobile ROS Platform*. (n.d.). s. (March 6, 2021), Trossen Robotic, https://www.trossenrobotics.com/interbotix-turtlebot-2i-mobile-ros-platform.aspx
8) *turtlebot_navigation/Tutorials/Autonomously navigate in a known map - ROS Wiki*. (n.d.). ROS.Org. (March 6, 2021), http://wiki.ros.org/turtlebot_navigation/Tutorials/Autonomously%20navigate%20in%20a%20known%20map
9) *find_object_2d - ROS Wiki. (n.d.),* (March 6, 2021), ROS.Org, http://wiki.ros.org/find_object_2d
10) I. (n.d.). *introlab/find-object*. GitHub.(March 6, 2021), https://github.com/introlab/find-object/blob/master/launch/find_object_3d_kinect2.launch
11) *Indoor mapping using kinect and ROS. (2015, August 1). IEEE Conference Publication. https://ieeexplore.ieee.org/abstract/document/7379780*
12) Crowe, S. (2018, February 5). "*TurtleBot 2i: New Chassis, Robot Arm Support, Intel Joule",* Robotics Business Review, https://www.roboticsbusinessreview.com/rbr/meet_turtlebot_2i/