Multi-Sensor Robot Platform Development for Odor Source Localization Task

Abstract

The paper discusses customizing a popular robot development platform "Turtlebot3" for Odor Source Localization (OSL) task. OSL technology allows a robot to locate and navigate to odor sources in an unknown environment. Turtlebot3 is an agile robot platform that includes Raspberry Pi for on-device computation and an OpenCR board for additional sensor connection. It utilizes runs on “Robot Operating System” that allows it to run complex algorithms that can subscribe and publish to specific robot sensor and components. In combination, this robotics platform can be customized to perform a wide varieties of robot tasks. This paper focuses the additional sensor installation for OSL experiment. It also includes an olfactory-based navigation algorithm named moth-inspired algorithm. The algorithm was applied on real-world experiment with varying conditions. The experiments show that moth-inspired algorithm successfully navigates to the odor source in laminar airflow environments. The paper also discusses future scope of adding vision sensors and machine learning algorithms.

Key words: Odor source localization, moth-inspired algorithm, turtlebot3, Robot Operating System, Multi-sensory robotics.

1. Introduction:

Humans and animals interact with external environment using sensory systems – visual, auditory, olfactory, gustatory, tactile, etc. These systems help animals sense and interpret the environment and perform activities like foraging, mating, evading predators, etc. for survival. A similar strategy of using multiple-sensors is also used in robotics for sensing and acting in unknown environments. A robot equipped with visual sensor (camera), olfactory sensor (e.g., chemical sensor), tactile sensor (e.g., touch sensor, airflow detection sensor), etc. can sense, navigate and manipulate unknown environments to achieve specific goals.

Olfaction is an important sensing system for robotics. Odor Source Localization deals with technologies that can allow robots to perform tasks such as detecting and navigating towards a target odor source in an unknown environment \cite{kowadlo2008robot}. OSL has increasingly important applications including monitoring air pollution \cite{dunbabin2012robots}, locating chemical gas leaks \cite{soldan2012robogasinspector}, locating unexploded mines and bombs \cite{russell2004robotic}, and marine surveys such as finding hydrothermal vents \cite{ferri2008novel}, etc.

Turtlebot is a popular and agile robotics development platform (https://github.com/RobinAmsters/GT\_mobile\_robotics). This paper utilizes the Turtlebot3 waffle-pi platform. The robot model is pre-built with camera, 360-degree LiDAR, gyroscope, accelerometer and magnetometer for sensing and DYNAMIXEL driver for navigating in new environments. It uses Raspberry pi 4 as central processing unit that allows complex on-board computation and easier communication with remote computers. It uses OpenCR (Open-source Control module for ROS) board as a powerful and customizable robot controller. The modularity of turtlebot makes it possible to customize it with additional sensors. Additionally, Robot Operating System allows minute control over specific robot components, so that it is possible to subscribe to specific sensors and publish to navigation from the remote PC. ROS supports both Python and C++ as programming language. Thus, existing mathematical libraries published in Python and C++ can easily be incorporated in algorithms written for the robot.

In addition to hardware and sensors, effective algorithm is required for Odor Source Localization task. It’s important to plan navigation based on changing odor plume and airflow data to reach odor source. Olfactory-based navigation algorithms achieves OSL goals by using detected odor plume and airflow direction/speed as cues to guide the robot in finding odor source \cite{farrell2002filament}. In this paper, a bio-inspired algorithm has been proposed for OSL task.

A bio-inspired algorithm makes the robot mimic animal odor search behaviors. Moth-inspired algorithm is a bio-inspired method which imitates male months mate-seeking behaviors \cite{carde1997mechanisms}: a male moth flies across the wind when pheromone plumes are absent and upwind when detecting plumes. This behavior can be framed as a `surge/casting' model \cite{lopez2011moth}, where a plume tracing robot traverses wind when missing plume contact (i.e., `casting') and moves against the wind direction when detecting plumes (i.e., `surge').

Contributions of this work can be summarized as: 1) discuss customization of the turtlebot3 robot platform for odor source localization experiments; 2) find search performance of moth-inspired algorithm in different real-world search environments; 3) discuss possibilities of combining more sensors and machine learning based algorithms in the same robotics platform.

The robotic platform and experiment field are presented in Fig. \ref{fig:demonstration}. The turtlebot3 robot was customized for OSL task, the search area included an odor source with fans. The location of the odor source was unknown to the robot and changed in different experimental runs. In the remaining of this paper, Section \ref{sec:review} reviews the recent progress of olfactory-based navigation algorithms; Section \ref{sec:method} reviews technical details of the robot customization and moth-inspired navigation algorithms; Section \ref{sec:experiments} presents experiment details of performing the real-world experiments.

1. Related works:

* Multi-sensor robot platform development.

1. Methodology:

* Platform development:
  + Turtlebot3 Waffle-Pi:
    - Platform: Turtlebot3 is a popular mobile robot system for research and education. It is highly modular and customizable.
    - ROS: Turtlebot3 uses Robot Operating System (ROS) as its operating system.
    - Raspberry pi
    - OPENCR
    - Given sensors: Turtlebot3 includes Raspberry Pi Camera, 360-degree LiDAR sensor, 3-axis gyroscope, 3-axis accelerometer, 3-axis magnetometer. These sensors help Turtlebot3 to measure 9-axis inertia.
    - Capabilities: Turtlebot3 has can perform SLAM (simultaneous localization and mapping), Navigation and manipulation tasks with the given sensors and DYNAMIXEL diver. It can be paired with additional sensors for increasing its functionalities.
  + System setup
    - Remote PC – description and role
      * A remote PC can be used to run custom programs for controlling the Turtlebot3. Turtlebot3 runs on ROS. Both Python and C++ is supported in this platform.
    - SBC, OPENCR – description and role
    - Included sensors: laser, odometry, camera – sensor description and assembly
    - Networking
  + Additional sensor assembly – highlight of the paper:
    - Sensor connection: for odor source localization, additional chemical sensor, airflow speed, wind direction sensors were needed.
      * Chemical sensor –introduction, how it works, connection type, connecting to OPENCR:
        + Introduction: MQ3 sensor is a widely used Metal Oxide Semiconductor (MOS) sensor. It operates on 5V DC and consumes about 800mW. It can detect alcohol concentrations ranging from 25 to 500 ppm.
        + Connecting to OPENCR: its VCC pin connects to 5V output (ping number ? of OpenCR). Its GND connects to pin number ? of OpenCR. Digital/analog pin connection.
      * Airflow and windspeed sensor:
        + Introduction
        + Connecting to OPENCR:
    - Sensor code: Turtlebot3 OPENCR firmware supports additional two touch detecting Bumper sensors, cliff detecting IR sensor, ultrasonic sensor, illumination detecting LDR sensor, etc. However, adding different sensors than the ones listed above require changing the OpenCR firmware code.
      * Chemical sensor edit:
      * Windspeed sensor edit:
      * Wind direction sensor edit:

Uploading changed OPENCR firmware:

* Odor source localization
  + Description of an OSL task: An OSL task have three stages – plume finding, plume tracing and source declaration [source 38]. The first stage is plume finding, which aims to detect plume in the search area. Once the robot detects plume in the search area, the plume tracing stage initiates. In this stage the robot uses the plume detection and airflow state to approach the odor source. Once it reaches the vicinity of the source, it records the source location and terminates.
  + Review of the olfactory-based moth-inspired navigation algorithm: the moth inspired method has two behaviors – moving in cross-wind trajectory to detect plume in the air, and if plume is detected, moving in up-wind trajectory to find odor source. If the robot loses plume in the air, it’ll continue to move in cross-wind trajectory to find plume again.

1. Experiments
   1. Experiment setup: Experiments were conducted in the Automatic Control Lab at the Louisiana Tech University. The lab area was divided into a search area where the robot can navigate and an operation area for the remote PC. The size of the search area is 27X11ft^2. The robot, odor and airflow source were randomly placed in this search area for each trial run. Ethanol vapor was employed as the odor source as it is minimally toxic and commonly implemented in OSL research [40]. A humidifier was used to disperse ethanol vapor consistently as the odor plume. An electric fan was used behind the humidifier to increase odor propagation.
   2. Mapping the experiment area
   3. Sensory input
   4. Experiment run: Fig \*\* depicts the search area and the mobile robot. During an experiment run, the robot sends sensor measurements to the remote PC. The remote PC runs the moth-inspired navigation algorithm to calculate robot’s heading command and transmit it back to the robot. The robot again collects and transmits sensor readings back to the remote PC. This cycle is repeated until the robot gets within 0.2m (the radius is determined based on the search area and robot dimensions) radius of the odor source.
   5. Moth-inspired algorithm
   6. Multiple runs: To evaluate the performance of the moth-inspired algorithm, total 6 tests were conducted. In each run, the odor source was initiated at a separate position.
   7. Description of a specific run
2. Conclusion and future works

* Sensor update: olfaction and vision integration with several paragraphs
* ML based methods: how to process sensor data with machine learning models to guide OSL.

References