WPI Summer Opportunities in Aerospace Research (SOAR)

Project:

Estimation of Spaciotemporal Fields with Mobile Sensing Robot

Supervisors:

Dr. Michael Demetriou and Dr. Nikolaos Gatsonis, Aerospace Engineering Department, WPI

Student:

David Acuna, Aerospace/Robotics Engineering, WPI

Period:

Summer 2024 (8 weeks)

Project Overview

This research project aimed to develop a platform for conducting real-world experiments with a simulated dynamic threat field based on solutions to a 2D advection-diffusion field. An array of projectors was set up to project a composite image of the threat field onto the laboratory testing grounds. This design allows for the creation and testing of various dynamic fields, where the colors of the projected image directly correlate to the gas concentration of the solution of the field. A mobile robot is then equipped with four camera systems that are capable of measuring and interpreting concentration readings from the color images taken which can then be used for targeted navigation or derivation of the original parameters of the threat field. This document contains detailed information about the various subsystems used during experimentation and the work done to them during the fellowship period.

Threat Field Image Generation

The images representing the threat field are created in a Dell Latitude 7330 Rugged Extreme Laptop that serves as a base station for all the heavy computing needed for simulation and navigation. They are generated using a script calculating a 2D solution to the advection-diffusion equations for a fluid field. The iterates over a 1920x1080 image and calculates the concentration of the field at any given point based on the initial parameters describing the position, strength, and movement of the source within the field. This concentration takes the form of a normalized value between 0 and 1 for every pixel with the image array. These values are then mapped onto RGB with the red part of the spectrum representing higher concentrations and violet purple representing lower concentrations, the values are then taken to produce a PNG image representing the computed field. This process is repeated every 5 seconds with a different time step to show the progression of the field over time.

After generating the visualization of the thread field, some additional processing is required to display the field on the projectors. The full image is divided into four quadrants and each component image is saved with a different filename and sent directly to the Raspberry Pis mounted to the projectors. The images for the top quadrants (projectors 1 and 2) are flipped 180 degrees because they will be projected from the opposite side of the workspace as the bottom quadrants (projectors 3 and 4).

As part of the project, it was my responsibility to change the previous generation of the threat field from a simple bivariant normal distribution to a dynamic model based on the solution to the advection-diffusion equation of fluid dynamics. This process required updating the current threat field generation systems with the new algorithm, furthermore, it requires changes of the previous file-sharing system of communication with the projectors from an NFS file-sharing system to a more direct file-sending protocol that would not have many problems with constant reads and writes of files due to the dynamic field updates.

Projector System

The projector system comprises four Optoma ZH450ST projectors which work are interconnected through a local network to display one large composite image onto the floor in the laboratory test area. The projectors are mounted onto large camera tripods arranged around the perimeter of the workspace facing downwards at a 90-degree angle. The protectors are numbered 1 through four and are placed in a counterclockwise order in the field. The Optoma ZH450ST projectors have a resolution of 1920x1080, a throw distance of 1.3' - 10.5', and an aspect ratio of 16:9.They are placed in 3 DOF mounts that can be easily adjusted to align the projectors properly to display an unwrapped image of the original threat field.

Each Projector is controlled by a Raspberry Pi 4 model B connected via HDMI. The Raspberry Pis are mounted along a leg of the tripods with 3d printed clamp mounts, which also hold the power supply and extension cords required to power the system. The Raspberry Pis are intended for headless operation and are controlled remotely from a laptop that serves as the base station. They must be powered via the USB-C port.

The system works by generating a color image of the threat field on the base station laptop based on the needed initial parameter of the experiments, splitting the images generated into 4 separate files using a Python script that corresponds to the four quadrants covered by the projectors. Then using the SCP protocol, it sends the images directly to the Raspberry PI overwriting the previously displayed image. The Raspberry PIs each run a script that searches for the image in its file systems and displays it in Fullscreen on the projectors with the FEH image viewer application. The PIs will continually check the folder and refresh the display when a new image is uploaded to allow for dynamically changing fields.

As part of the project, it was my responsibility to change the previous projector system using a different Optoma projector from the current one. This required a redesign of CAD models for all the mounting systems described above, as well as manufacturing of the new models using FDM 3D printers. Furthermore, it required extensive recalibration of the systems to have the most optimal parameters for experimentation and have the least amount of warping and deformation from the four different stitched images. Furthermore, this required an update to the system mounted on the Raspberry Pis on the projectors to use the new network file-sharing protocol.

Camera System

Hardware: The Camera System is comprised of a Raspberry PI 4 mounted on top of the mobile robot connected to four Logitech C920x webcams. The system is powered by a portable power bank which powers the Raspberry PI via the USB-C port. The two portable power banks in the lab are interchangeable so one can charge while the other is in use.

The cameras are connected to the USB-A ports on the Raspberry PI and should be automatically recognized by the system. Afterward, the camera settings such as white balancing and autofocus are edited using the v4l2Linux program. The webcams are secured to the Turtlebot4 with 3D-printed mounts that attach to the screw holes on the robot chassis intended for the standoffs for the optional mounting plate. The Raspberry PI for the camera system is also mounted on this structure. The camera mounts were designed such that the four cameras are arranged in a radially symmetric pattern about the robot, each an equal distance from the center point directed at 15-degree angles and lowered to a height of 50mm from the top of the robot. The cameras are numbered 1 through 4 and correspond to the direction of the robot for navigation.

Camera System ROS node: The Raspberry PI mounted on the camera system is running the ubuntu operating system with ROS or robot operating system for interconnection with the laptop base station as well as the mobile robot itself. This system runs a ROS node called "cam_sys" which contains a ROS service called "get_color_vals" that when called accesses the cameras attached, and captures an image using the OpenCV library, The script then computes the average RGB values for a square of pixel values located at the center of the image. The RGB values are then converted to HSV, from which the hue value is extracted and normalized between 0 and 1. This is effectively an inverse of the encoding process used to generate the image of the threat field. The process is repeated for all four cameras, and the values are returned to the node that made the service call.

As part of the project, it was my responsibility to change the previous camera system to use the new Logitech C920x webcams which were changed due to their ability to be manually configured and optimized to get the best color readings. This is due to the fact that the previous cameras had a problem with automatic white balancing where some shades of yellow where balanced to whit causing a misreading of colors when traversing the threat field in the experiment.

Mobile Robot

TurtleBot 4: This project makes use of the Turtlebot4 mobile robot platform. The Turtlebot4 is a two-wheel, differential-drive robot. The robot has two onboard computers which are a Raspberry Pi 4 and a Create3. These devices must both be connected to the Wi-Fi network to operate the robot. The Turtlebot4 uses ROS2 Humble to communicate with the rugged laptop. The Turtlebot4 automatically launches the bring-up node automatically upon power up, so the user does not need to run any nodes on the robot manually. The robot communicates with remote devices over ROS topics to report information from sensors and to receive actuation commands. The Turtlebot4 may be teleoperated with the game controller joysticks as explained in the TurtleBot documentation.

ROS Platform: The robot is controlled by running the ROS node on the laptop base station, the ROS workspace in the base station has two node functionalities, one that calls the service in the camera system to get the current values of the cameras and another to navigate the threat field using these vales. The first node works by calling the ROS service from the camera systems that request the current color values at the center of the cameras, this message is then sent over the network and received by the laptop which then computes the normalized concentration value based on the hue reading. It then publishes these values to a topic which is then read by the second ROS node called "threat_field_naviator".

The goal of the threat field navigator node is to direct the robot to the maximum of the threat field. This is accomplished by using data from the camera system to essentially measure the local gradient at the robot's current position in the field. From the four camera readings, the program selects the two highest readings and performs linear interpolation to compute a new heading direction in the direction that the value of the field is increasing. The robot then performs a point turn to orient itself in this direction and drives forward for a short distance. This process is repeated such that the robot will make a service call to the camera system to sense the field, perform some computation, and then execute a movement to get closer to the goal.

Simulated Environment

Lastly, there is the fully simulated environment component of the experiment. The objective of this full simulation was to test the viability of the underlying navigation and sensor reading components of the system. Through this simulation we were able to verify that the navigation algorithm of the threat field is both accurate and reliable in any different positions and initial conditions, therefore most improvements that can be made would be in terms of the physical hardware of the experiment itself. The simulation runs in a ROS-based physics simulator called Gazebo where a virtual test environment was set up with a static projected field as the mesh for a ground plane, which would then be read by virtual cameras mounted upon a model of the TurtleBot 4s used in the experiment itself. With this configuration of the simulation, it is not only possible to test the fully virtual behavior of the system but also to visualize the response of the physical experiment due to the interconnection of the ROS nodes being identical for both the simulation and the actual robot. This portion of the project was fully my addition and was part of my responsibility during the duration of the fellowship.

Project Results

The result of the fellowship was a complete and reliable platform for conducting real-world experiments with a simulated dynamic threat field based on solutions to a 2D advection-diffusion field. With this system, it is possible to expand the experiment to then be able to use the readings from the mobile robot to predict the parameters of the original threat field while navigating thought it, which could help determine parameters such as origin of release and concentration of threat agent.