Pathfinding Algorithm development and Live Data Simulation Integration for Enhancing Safety Systems in Vehicles

Jonathan Randall, David Santiago Obando Ortegon and Levi McKinney
Faculty Mentor: Dr. Valentin Soloiu, Professor, Intelligent Vehicles Laboratory
Georgia Southern University, Department of Mechanical Engineering

PRESENTED AT

BACKGROUND

1) Develop algorithms to allow for real-time data acquisition, simulation, and computation.
2) Process data from the environment to plan the best possible route to reach a destination.
3) Potential applications: autonomous vehicles, exploration, reconnaissance, manufacturing, simplified overview of larger systems.

RESEARCH HYPOTHESIS

Given a vehicle with various imaging, proximity, and position sensors, algorithms can be made to efficiently control the vehicle with intelligent decision making and reliable pathfinding while simultaneously displaying useful information to the operator.

LITERATURE REVIEW

Given a vehicle with various imaging, proximity, and position sensors, algorithms can be made to efficiently control the vehicle with intelligent decision making and reliable pathfinding while simultaneously displaying useful information to the operator.

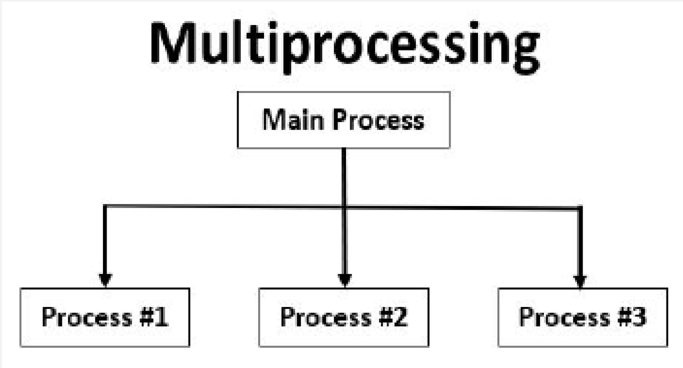


Fig. 1 - Multiprocessing splits up the processes to separate cores, allowing the CPU to divide and conquer.

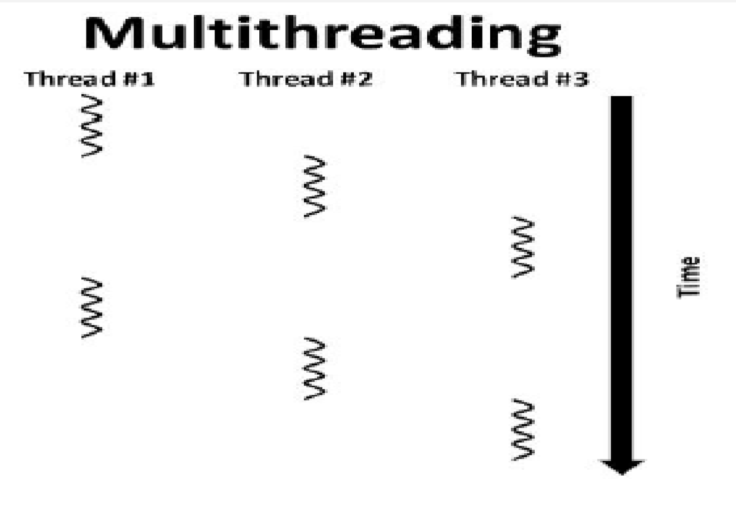


Fig. 2 - Multithreading uses all the cores in the CPU at once, jumping between processes quickly to create a pseudo-parallelism.

Noise filtering is the process of using an algorithm to reduce statistical noise and outliers from the data to keep the data as true as possible. The Kalman filter is a two-step algorithm commonly used in navigation and control of vehicles. The algorithm was chosen as it can be used in real time with the sensor, requiring only the current data and the previous prediction with its uncertainties. Figure 3 shows how it works.

Pathfinding is the computation of the most efficient route between two given points. Dijkstra's algorithm is the most well-known pathfinding algorithms. This algorithm uses a brute force like approach to find the best possible path. A* pathfinding is another algorithm that works similarly to Dijkstra's, but it examines every neighboring node for a 'cost'. For our purposes, A* was chosen as the complete map would never be known, and quick solutions should be prioritized over the absolute best solution as the pathfinding algorithm will be run repeatedly as the vehicle is travelling. Figure 4 shows A* pathfinding on the left, and Dijkstra's on the right.

METHODOLOGY

Berry IMU v3: Accelerometer
HW-201: Infrared Sensor
RpLidar A3: Lidar Sensor
HC-SR04: Ultrasonic Sensor

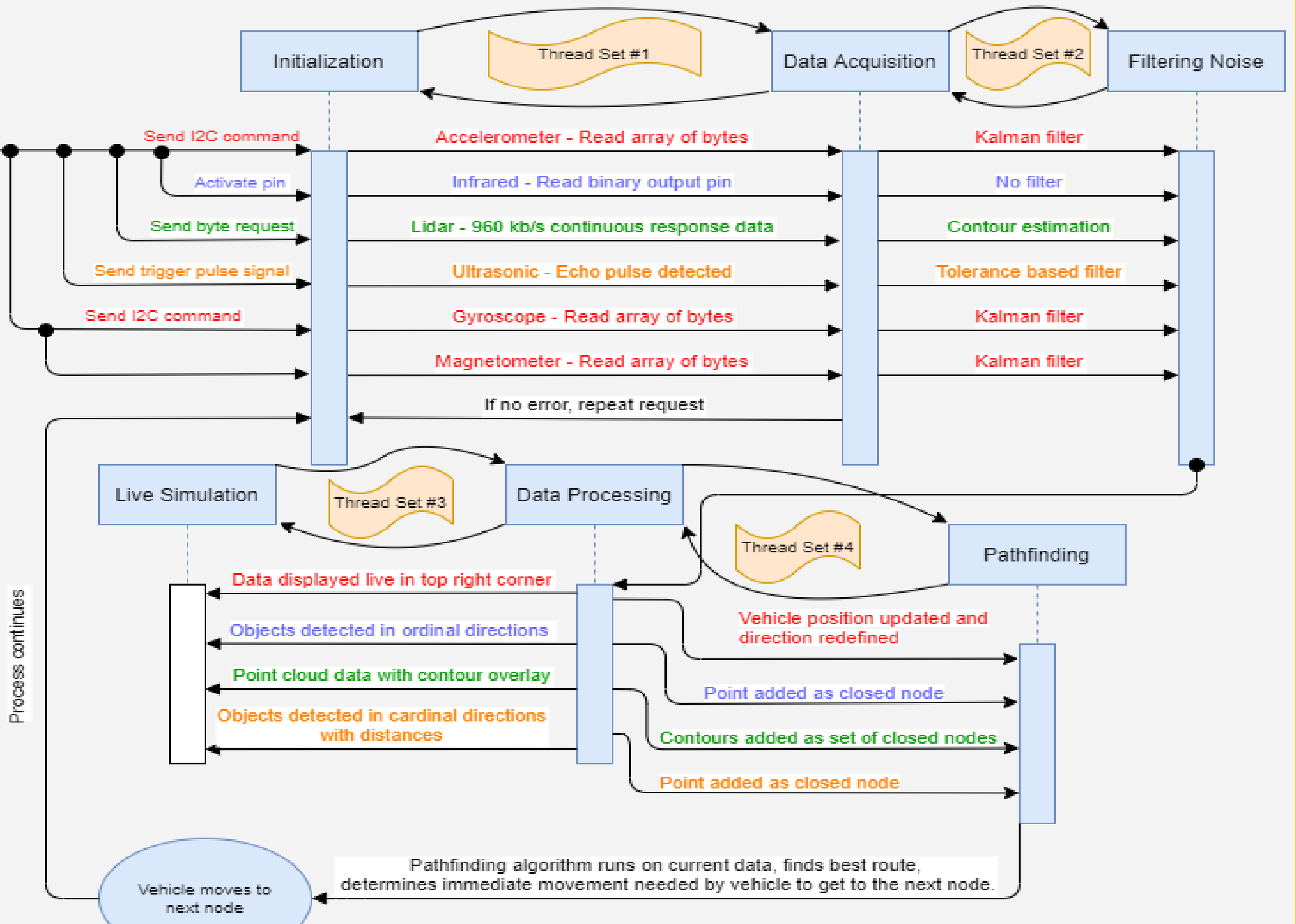


Fig. 3 -The process starts by sending request to all the sensors.

Data is acquired from the sensors and filtered to increase accuracy.

Data is sent to the live simulation as well as the pathfinding algorithm concurrently for operator visuals as well as autonomous control.

RESULTS AND DISCUSSION

1) Pathfinding algorithms proved to be effective at navigating indoor and outdoor environments, allowing for autonomous travel.
2) Pathfinding with no prior knowledge of the environment allows for use in exploratory situations, as well as for routine routes with unexpected obstacles / environmental forces.
3) This will lead to safer transportation of people and goods, removing human error other variables introduced with human drivers.
4) Live simulation of the sensor data allows the operator to keep eyes on the surrounding environment and take over if necessary.
5) Noise filters on all incoming data allowed higher quality information to be extracted from lower quality sensors, thus saving money.

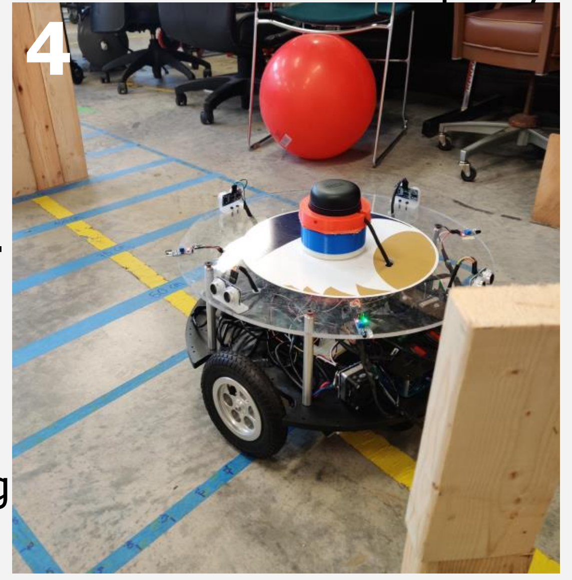
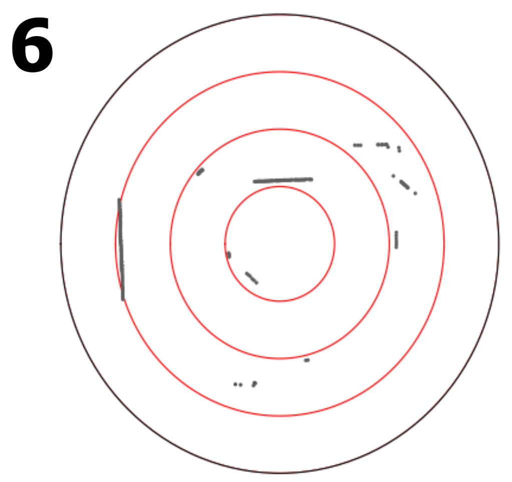
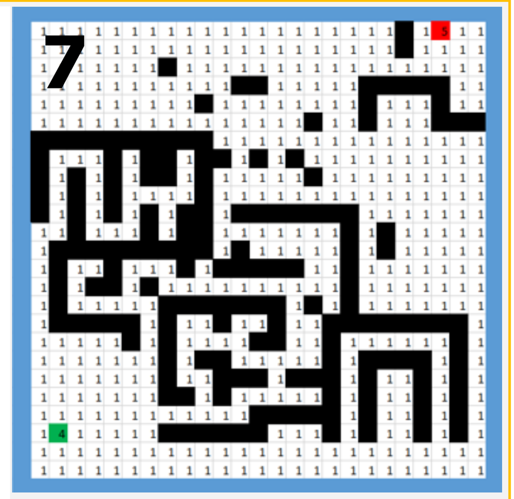


Fig. 4 – Vehicle pictured while traveling through an indoor environment. Notably seen are obstacles in front, on the back left, and to either side, and red ball in the front right.
Fig. 5 - Live simulation of data displayed on a tablet at the instance pictured. Notice 4 main obstacles being detected by ultrasonic/infrared sensors. The blue line on the graph is the path the vehicle is currently planning on navigating based on the known environment.
Fig. 6 - Live simulation of LIDAR on second tab of tablet. This allows for a better estimation of the size of obstacles, as well as picking up on surrounding in all 360 degrees, rather than just the cardinal and ordinal directions.





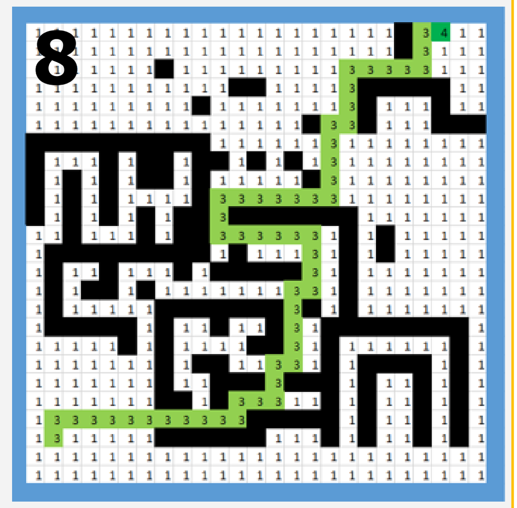


Fig. 7 – Excel based prototype of pathfinding algorithm. Predefined environment shown on left with the green cell being the start point and the red cell being the end. In the actual use of the autonomous program, the environment starts completely blank with a goal and a start point, and the obstacles are detected with sensors in real time as the vehicle progresses towards the goal.
Fig. 8 - Same prototype as shown in Figure 9 after applying the pathfinding algorithm. The light green cells show the path traveled. Computational for this 25x25 matrix took approximately 0.24ms on a i7 processor, and 0.83ms on the Raspberry Pi 4. This gave us an idea on how to divide nodes based on real life distance given our CPU.

DISCUSSION

1) Due to the limited quad-core capability of the Raspberry Pi 4 Model B, the CPU can only handle running up to 4 processes at once without the need for multithreading. This limits the effectiveness and speed of the algorithms severely. Better processing power would allow for more refined data to be used for more accurate pathfinding and simulation.
2) Live data simulation could be further fine tuned to make a more aesthetically pleasing and user-friendly interface in the future, with auditory alerts included as well to keep the operator in tune with the surroundings more effectively.
3) Higher grade sensors would lessen the need for additional noise filters, allowing for more accurate data to be acquired and fed into the various algorithms. With the existing sensors various issues were discovered that could not be fixed through programming alone.

TOP FINDINGS

- Through use of noise filters, and comparing multiple sensors against each other, high quality information can be obtained through low quality sensors, thus saving resources.
- Pathfinding algorithms that can develop working maps from scratch through use of sensor data allow for route planning in new, unexplored environments as well as routine trips with unexpected conditions.
- Potential applications include safer self-driving manned vehicles that can navigate with less reliance on GPS, UAVs that respond to quick changing environments, exploratory vehicles on Earth and beyond, unmanned reconnaissance and rescue, dynamic manufacturing and delivery, and live simplified tracking of complex systems.

REFERENCES

[1] Sensor fusion using the Kalman filter. (n.d.). Retrieved March 22, 2021, from http://campar.in.tum.de/Chair/KalmanFilter
[2] A* (a STAR) Vs Dijkstra's Algorithm pathfinding grid visualization - JavaScript. (2018, April 06). Retrieved March 22, 2021, from https://www.youtube.com/watch?v=9REexHx0hDY
[3] The key to lidar safety. (2021, January 18). Retrieved March 29, 2021, from https://www.schott.com/innovation/en/the-key-to-lidar-safety/
[4] Neal, A. (2018, April 24). Lidar vs. radar. Retrieved March 22, 2021, from https://www.fierceelectronics.com/components/lidar-vs-radar
This research was supported by DoD-NSF Assure REU Site Award: 1950207