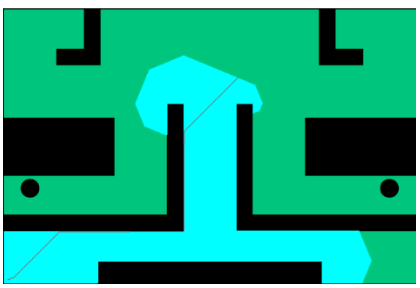
A Review of Search- Based Path Planning Algorithms for Indoor Navigation of UV Sterilization Robots

Results

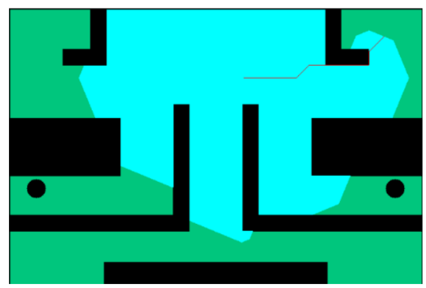
The tests were conducting by writing a code which justifies the algorithm suggested in Methodology. Let us look at the results step by step to see if basic search-based algorithms can be implemented for path planning in indoor environments.

Here, three contamination points were selected from entire map randomly. While implementing we figured out that there are good chances that the step 1, where we calculate the centroid of the plane formed by the contamination point might lie in obstacle space. To overcome it, a default point was set such that it is equal to the center of the map which does not lie in obstacle space and this point will be accessed by the algorithm if the calculated point is in obstacle space or not valid.

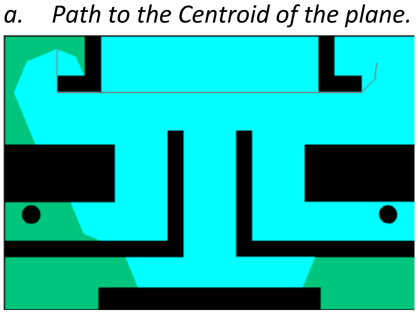
When the tests were conducted for Dijkstra algorithm, on an average total time taken for calculating the path for every point was 330 secs.



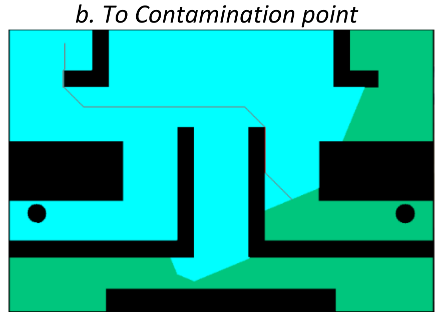
a. Path to the Centroid of the plane.



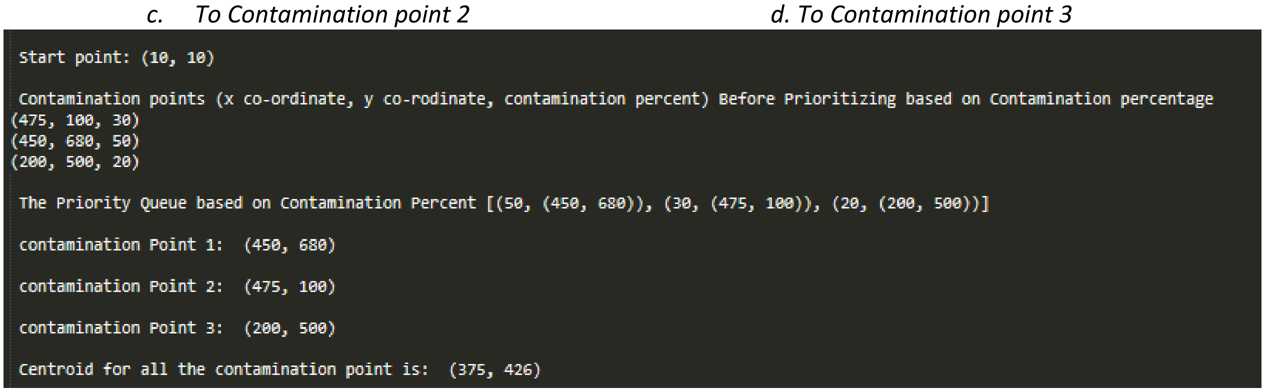
b. To Contamination point



c. To Contamination point 2



d. To Contamination point 3



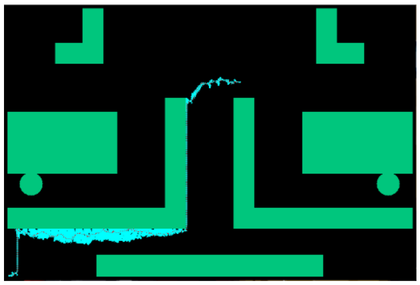
```
Start point: (10, 10)

Contamination points (x co-ordinate, y co-rodinate, contamination percent) Before Prioritizing based on Contamination percentage
(475, 100, 30)
(450, 680, 50)
(200, 500, 20)

The Priority Queue based on Contamination Percent [(50, (450, 680)), (30, (475, 100)), (20, (200, 500))]

contamination Point 1:  (450, 680)

contamination Point 2:  (475, 100)

contamination Point 3:  (200, 500)

Centroid for all the contamination point is:  (375, 426)
```

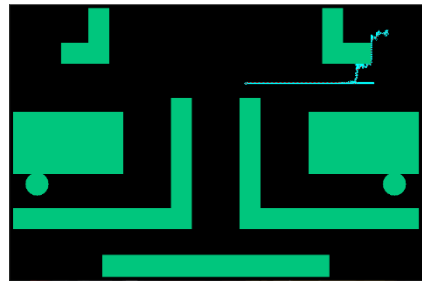
e. The calculations in algorithm.

Fig. 1 Results for path planning algorithm using Dijkstra

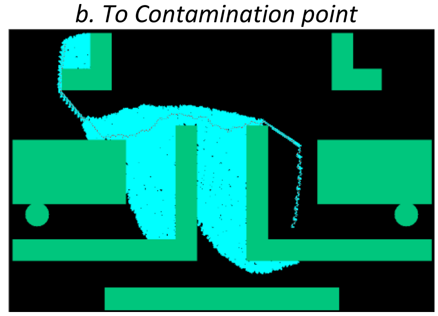
We can se how a priority queue was made based on the contaminated percentage assigned to every goal point on the map.

When the tests were conducted for A* algorithm, on an average total time taken for calculating the path for every point was 140 secs.

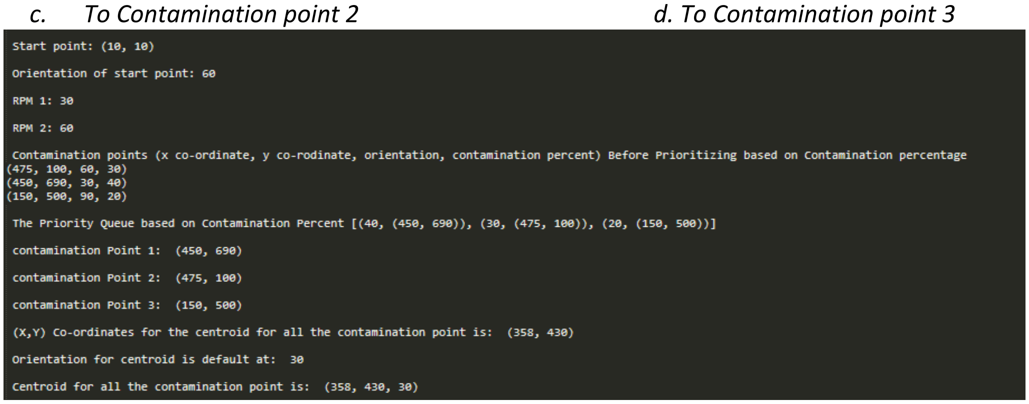




a. Path to the Centroid of the plane. *b. To Contamination point*



c. To Contamination point 2 *d. To Contamination point 3*



```
Start point: (10, 10)

Orientation of start point: 60

RPM 1: 30

RPM 2: 60

Contamination points (x co-ordinate, y co-rodinate, orientation, contamination percent) Before Prioritizing based on Contamination percentage
(475, 100, 60, 30)
(450, 690, 30, 40)
(150, 500, 90, 20)

The Priority Queue based on Contamination Percent [(40, (450, 690)), (30, (475, 100)), (20, (150, 500))]

contamination Point 1:  (450, 690)

contamination Point 2:  (475, 100)

contamination Point 3:  (150, 500)

(X,Y) Co-ordinates for the centroid for all the contamination point is:  (358, 430)

Orientation for centroid is default at:  30

Centroid for all the contamination point is:  (358, 430, 30)
```

e. The calculations in algorithm.

*Fig. 2 Results for path planning algorithm using A**

Let us compare both search-based algorithms for our case. (demonstrated in table 1)

Tests for:	Dijkstra	A*
Search Algorithm	Greedy Best First search	Best First Search
Time efficiency	Less	High
Number of explored nodes	High	Less
Usage of Heuristics Function	No	Yes
Time Complexity	$O(n \log n)$	$O(n^2)$

table 1. Comparison of observations and key features.

Apart from taking these factors into consideration, a lot of updates can be made in the way an entire UV Sterilization robot system is made. These algorithms can be integrated with Robot Control Software with multi-functional GUI and take out the load of manually guiding the robot at every step and reduce Human error.[3] By making the manual navigation secondary and converting that into more of human monitoring at most of the steps, increases the accuracy of system.

The requirements of the GUI will be as follows:

1. It is not only paired with the Robot but also with the sensor system. Where the sensor system will have calibrated cameras able to capture the dimensions of environment and account for the changes in the position of furniture (Beds, medical monitoring devices etc.) and Bio-Sensors capable of sensing contamination point specification (Position in room, percentage, kind of bacterial infestation etc.)
2. Can access overview of Map Environment before planning: This can be done by integrating the Sensor system installed throughout the environment.
3. Can access Contamination point information through sensor systems.

The disinfection capability of robot will vary based on how far it is from the contamination point [4], but based on the hardware specification (UV light used in robot), a manual radius can be set such that the path between contamination point can be further optimized. This helps us consider which algorithm should be implemented if we choose to use basic search-based path planning algorithm for navigation in UV sterilization robots. Also, a halt will be added when the robot actually reaches a contamination point such that robot waits for the time period which is equal to the exposure time required for the inactivation of infestation in that particular zone.[2] Based on the kind of infestation (bacteria), brightness of the light can also be modified to increase or reduce the exposure time.

Conclusion:

Dijkstra is the best uninformed search algorithm to reach the goal point, but it only considers cost required to come to the next node. As it expands in every direction equally, it increases speed required to reach goal node and the number of explored nodes. A* is much better fit in out situation as we can compute the cost to come and cost to goal i.e. heuristic function, and this can be calculated in our case. Although, information required to compute is less in Dijkstra than A*, the processing speed for A* is much faster than Dijkstra. Hence, we can see that both algorithms have their own unique importance. A* best fits the indoor environment as the case of infinite points is rare (as the indoor are will be finite), also heuristic approaches are close to human way of behavior learning which makes it more efficient if implemented in a robot system which have machine learning capabilities.[5]

This will help us understand how we can choose between the search algorithms while implementing them in UV sterilization robots for improving the navigation technique.

References

[1] Goyal, Jitin Kumar, and K.S. Nagla. *A New Approach of Path Planning for Mobile Robots - IEEE Conference Publication*, 2014, ieeexplore.ieee.org/document/6968200/.

[2] P. Chanprakon, T. Sae-Oung, T. Treebupachatsakul, P. Hannanta-Anan, and W. Piyawattanametha, "An Ultra-violet sterilization robot for disinfection," *2019 5th International Conference on Engineering, Applied Sciences and Technology (ICEAST)*, 2019.

[3] El Zant, Chowkis, et al. *UV-Robot Supervision System Design and Development*. July 2018, www.researchgate.net/profile/Mhammed_Sahnoun2.

[4] Rubæk, Thomas, et al. *Evaluation of the UV-Disinfection Robot*. Blue Ocean Robotics - for Humans, 2016, prhoinsa.com/images/pdf/uvd/UVDR-Whitepaper.pdf.

[5] Anantha, Sai Hari Haran, et al. *A Survey of Autonomous Mobile Robot Path Planning Algorithms*. *IEEE Conference Publication*, 2017, ieeexplore.ieee.org/document/8378228/.