Global Scientific JOURNALS
GSJ: Volume 10, Issue 6, June 2022, Online: ISSN 2320-9186
www.globalscientificjournal.com

Calculating A* route of Drones in an Octree Environment:

*Rohaan Advani (Students, College of Engineering Pune)

ABSTRACT

An Octree is a spatial data structure that helps create a 3D environment using a coordinate system. A* route is the most optimal root from a start point to an endpoint associated with minimum cost. This paper attempts to discuss the methodology of adding a system of drones to an octree-based environment and performing A* routing on them from one point to another.

Date of Submission: 20 - 06 -2022 Date of Acceptance: - -2022

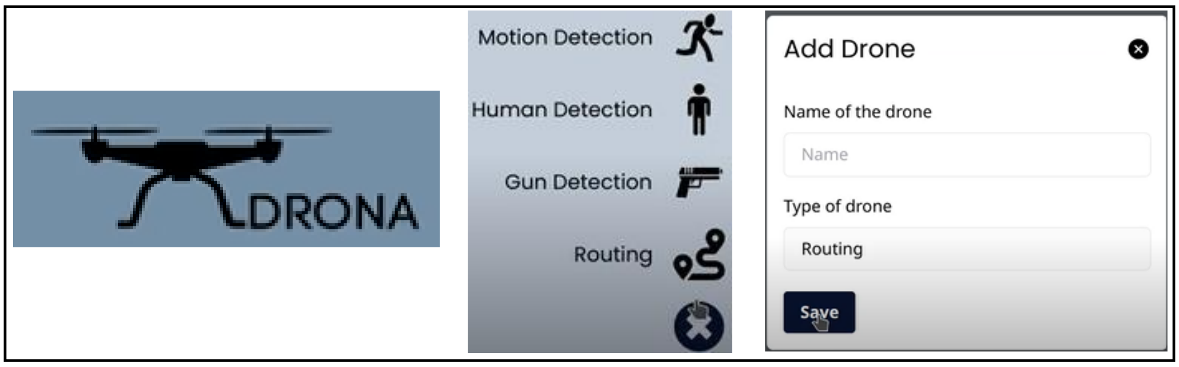
1. INTRODUCTION

An octree is a spatial data structure that stores a set of points with x, y, and z coordinates in a given coordinate system. This paper attempts to use this structure to build a 3-dimensional environment for simulating an optimal routing algorithm on drones. The concept discussed in the paper has been used to create a simulator software called Drona. Drona includes Adding, Removing, Editing, and Renaming Drone objects in an Octree Environment, Performing A* Routing on the Drones from one point to another in the given system, and a few Advanced Computer Vision applications such as Motion Detection, Gun Detection, and Human Body Detection which intend to simulate Threat Recognition and Hostile / Hostage missions for Military Applications of Drones.

Search algorithms are very popular in the field of artificial intelligence. There are multiple types of search algorithms that help determine an ideal route associated with a minimum cost of going from one point to another in a given coordinate environment. The different types of routing algorithms include Breadth-First Search, Depth First Search, Iterative Breadth Search, Iterative Depth Search, Uniform Cost Search, Greedy Search, and A* Search. The most optimal algorithm of these is the A* search algorithm which searches for the best route with minimum complexity.

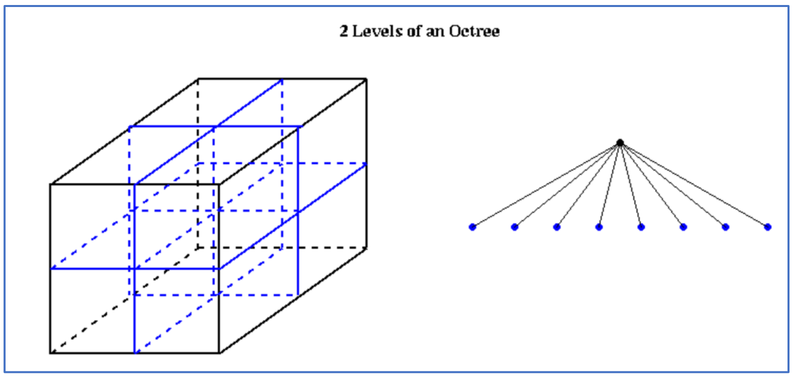
Project Domain – Drone Technology, General Public Services, Emergency, and Relief Services.

Applications – military, security, agriculture, transport, delivery, industry, remote sensing, etc.



2. IMPORTANT TERMINOLOGY

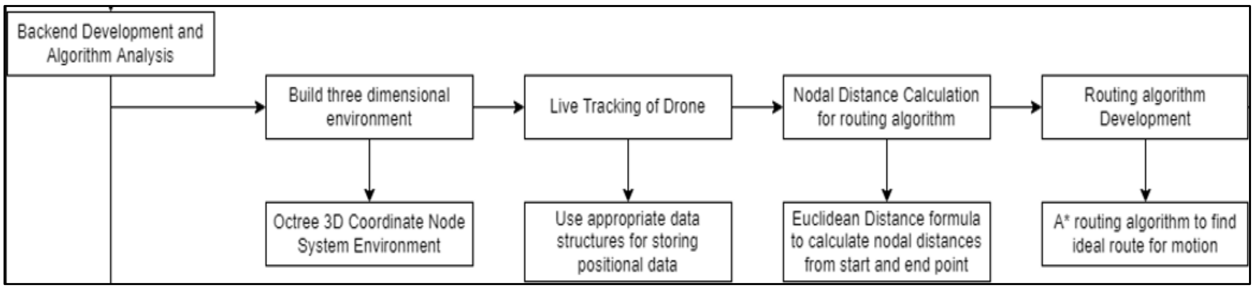
- Octree - An octree is a tree data structure in which each internal node has exactly eight children. Octrees are most often used to partition a three-dimensional space by recursively subdividing it into eight octants. Octrees are the three-dimensional analogy of quadtrees.



- Optimal Route - Route optimization is the process of determining the most cost-efficient route. It's more complex than simply finding the shortest path between two points. It needs to include all relevant factors, such as the number and location of all the required stops on the route, as well as time windows for deliveries.
- A* routing algorithm - A* is an informed search algorithm, or a best-first search, meaning that it is formulated in terms of weighted graphs: starting from a specific starting node of a graph, it aims to find a path to the given goal node having the smallest cost (least distance traveled, shortest time, etc.).
- Initial State – In the case of routing the initial state also known as the start state can be defined as the coordinates where the object is currently located in an environment. It is represented as the Initial state (s_0).
- State Space – Set of possible points the object can be in the environment State -> {s_0, s_1, s_2, ..., s_n}. The state-space can be navigated using the actions.
- Actions – Refers to the set of actions the search algorithm can take at a given state to find an optimal route. It is represented as Actions (s) -> {a_1, a_2, a_3, ..., a_n}. In the case of routing action is the motion of an object from one state to another.
- Result – Refers to the new state that the object will reach if action a_i is applied on the object when it is in state s_j where i, j is between 1 and n - Result (s, a) -> s'.
- Step Cost - Refers to the cost incurred in taking an action. In the case of route finding, it can be the distance traveled while going from one node to another.
- Path Cost -It is the summation of individual step costs along a path consisting of one or more nodes.
- G-score – It is a routing algorithm metric used which is the path cost of a node from the Initial State to the Current state.
- H or Heuristic – It is used to estimate the distance between a node and the goal. In this case, we'll be using Euclidian distance between two nodes as the heuristic.
- F-score – It is a routing algorithm metric used to determine how fruitful moving to the neighbor will be in finding the optimal path - F = G + H.
- Drone Objects – For the purpose of this project drones will be added to the environment as an object with the attribute as the current position which should ideally change after the routing procedure is complete from one point to another in a 3-Dimensional environment.

3. DESIGN MODEL

Given Below is the design model for the project:


Backend Development and Algorithm Analysis
Build three dimensional environment
Live Tracking of Drone
Nodal Distance Calculation for routing algorithm
Routing algorithm Development
Octree 3D Coordinate Node System Environment
Use appropriate data structures for storing positional data
Euclidean Distance formula to calculate nodal distances from start and end point
A* routing algorithm to find ideal route for motion

1. Build 3D Environment – Octree Data Structure
2. Live Tracking of Drone – Store Current Position
3. Nodal Distance Calculation – Calculate G, H, and F scores of nodes
4. Routing algorithm – Save A* route and test validity

4. PROPOSED METHOD

Following are the steps followed to get the optimal route for a drone to move in an octree environment and validate the route:

STEP 1: CREATE AN OCTREE NODE STRUCTURE

The Octree node should contain the following attributes to serve the purpose of building a 3D Environment:

1. Position – X / Y / Z coordinates
2. Size – Used for setting bounds (Integer)
3. Depth – Depth of a particular node in a tree (Integer)
4. IsLeafNode – Indicates whether the node is a leaf node or not (Boolean)
5. Branches – List of 8 Branches with respect to 8 Octants of a particular node
6. Lower Bound X / Y / Z = Position – (Size/2)
7. Upper Bound X / Y / Z = Position + (Size/2)

STEP 2: CREATE CLASS OCTREE AND OCTREE FUNCTIONALITY

Following is the Octree and A* functionality created:

INIT: Initializes an octree node with default values:

1. Sets Root Node Position as origin (0, 0, 0)
2. Sets Size of root node as world size given as input by user/programmer
3. Sets max objects per cube after which cube subdivides itself
4. Sets Depth of root node as 0
5. Sets Branches of root node as Null
6. Calculates Lower and Upper Bounds with respect to world size

INSERT: Insert a new node into the tree in the appropriate position:

1. Check if the position of the new node is within bounds of world size
2. Case 1: Build Node in Appropriate Branch if Branch Empty
    a. Calculate the appropriate branch using the find branch function
    b. Set New Position as per branch chosen

   c. Return Octree node with the new position and incremented depth by 1
3. Case 2: We are still at an internal node and required to go to the appropriate branch. Find the appropriate branch using the find branch function and insert a node in that branch
4. Case 3: Node is a leaf node with objects already in it
   a. If the number of data points after inserting a new node will not exceed the limit set in init then the node can directly be added.
   b. Else there is a requirement to split that branch in octants then choose the appropriate sub-branch with the find branch function and insert a node.

FIND BRANCH: Function to find the appropriate branch to insert

1. If X of node > = X of parent node we are comparing to increment branch by 4
2. If Y of node > = Y of parent node we are comparing to increment branch by 2
3. If Z of node > = Z of parent node we are comparing to increment branch by 1
4. After these conditions, the branch variable will range from 0 to 7 each indicating a respective octant which is returned to the insert node function to insert in the appropriate branch number.

CALCULATE G SCORE: Function takes user input of start point and calculates G scores of all other points added to the environment. G score is the Euclidean Distance of each point in the environment from the start point. G Scores are used during the uniform cost search algorithm which attempts to choose a route that has the least cost with respect to the start point. The recursive function sets G scores of all sub-branch nodes also.

CALCULATE H SCORE: Function takes user input of endpoint and calculates H scores of all other points added to the environment. H score is the Euclidean Distance of each point in the environment from the endpoint. H Scores are used during the greedy search algorithm which attempts to choose the route which reaches the endpoint the fastest and in turn chooses points with the least distance from the endpoint. The recursive function sets H scores of all sub-branch nodes also.

CALCULATE F SCORE: Calculates F scores of all points added to the environment. F score is the sum of all the G and H scores of all the points in the environment. Used mainly for the A* routing algorithm. The recursive function sets F scores of all sub-branch nodes also.

COLLECT: Function used to collect the node position and respective F score for all the nodes in the environment recursively adding them to a list. Later used for testing purposes.

COLLECT GHF: Function used to collect the node G score, node H score, and respective F score for all the nodes in the environment recursively adding them to a list. Later used for testing purposes.

STEP 3: BUILD TEST OBJECTS / OCTREE NODES FOR INSERTION

Following are the steps to build the 3D environment:

1. The structure of test objects should contain node position, G, H, and F scores
2. Set the Number of test objects, origin, and world size manually
3. Use the random function to create random test objects and add them to a list
4. Initialize Octree – Thus building a 3D environment
5. Iterate through the list of test objects and insert nodes into the tree

STEP 4: CALCULATE G, H, and F SCORES

Following are the steps to calculate the A* scores:

1. Get Start and End point of drone from user/programmer
2. Calculate G scores of all nodes using the start point and calculate G function
3. Calculate H scores of all nodes using the endpoint and calculate H function
4. Calculate F scores of all nodes using calculate F function

STEP 5: GET A* ROUTE AND SAVE TESTING FILES

Following are the steps to get the optimal route:

1. Call Collect Function storing position and F scores of all environment points into an array and write array content to env.txt file
2. Call Collect GHF Function storing G, H, F scores of all environment points into a different array and write array content to ghf.txt file
3. Sort the array with the environment points with respect to their F scores
4. Save the first 10 node positions in route.txt file as these are the points with the minimum F scores and these are the points that would provide an optimal route for the drone to travel

STEP 6: TESTING

1. The 3 files generated are now used for testing purposes
2. Route Validity Test – Checks if all points in route.txt file are available in env.txt file
3. Route Count Test – Checks if the number of points in route.txt file is 10
4. Environment Count Test – Checks if the number of points in env.txt file is equal to the number of test objects declared by the user which were inserted into the environment
5. A* Calculation Test – Checks if the G and H scores of all points sum up to the F scores respectively in the ghf.txt file.
6. If all tests print passed it indicates that the optimal route has successfully been calculated for the drone object.

5. FUTURE SCOPE

Drone technology and software will be very useful in the future when drone technology is going to be extended to applications like military, security, agriculture, remote sensing, delivery, etc.

The project software can be connected to sensors of an actual drone and the python script can be edited to handle real-world environments and obstacles. Moreover, features of computer vision can be implemented to make drone software standardized and useful for a variety of applications.

6. REFERENCES

[1] https://en.wikipedia.org/wiki/Octree/

[2] https://www.verizonconnect.com/glossary/what-is-route-optimization/

[3] http://www.ccpo.odu.edu/~klinck/Reprints/PDF/wikipediaNav2018.pdf

[4] A learning based algorithm for drone routing – Umut Ermağan, Barış Yıldız, F.Sibel Salman – Computers & Operations Research - Volume 137, January 2022, 105524.

[5] Algorithms for Routing an Unmanned Aerial Vehicle in the presence of Refueling Depots – Kaarthik Sundar, Sivakumar Rathinam.

[6] An Optimal Routing Algorithm for Unmanned Aerial Vehicles – Sooyeon Kim, Jae Hyun Kwak, Byoungryul Oh, Da-Han Lee, Duehee Lee.