Coverage Path Planning with Quadcopters

Ashwin Varghese Kuruttukulam*
Graduate Student, Robotics
University of Maryland, College Park
ashwinvk94@gmail.com

Mithun Bharadwaj*
Graduate Student, Robotics
University of Maryland, College Park
mithun02@umd.edu

Rachith Prakash*
Graduate Student, Robotics
University of Maryland, College Park
rprakas2@terpmail.umd.edu

Keywords: Coverage Planning, UAV, Path Planning, Cell Decomposition, A* search algorithm

Abstract— **The primary aim of this paper is to perform a simulation of optimal coverage path planning using quadcopters for a known environment with stationary obstacles using a suitable cellular decomposition technique. The performance metric used is the percentage of area covered. The target space is divided into cells such that there are no obstacles within each cell. This allows the quad-copter to perform incremental movements along each axis to cover area in each cell. Different planning techniques are used for cell to cell coverage and within cell coverage. To cover area within a cell, a simple back and forth movement of quadcopter is implemented. The path from one cell to the other is obtained through A* algorithm with collision avoidance.**

Finally, we validate the algorithm by simulating the quad-copter using the V-REP 3.4.0 simulation environment[1]. The simulation code is available at: https://github.com/ashwinvk94/vrep_quad_exploration

I. Introduction

Coverage Path Planning (CPP) is the task of determining a path that passes over all points of an area or volume of interest while avoiding obstacles. This task is integral to many robotic applications such as vacuum cleaning robots, lawnmowers, autonomous underwater robots creating image mosaics.

Quad-copters have shown to be useful for a variety of tasks ranging from landscape mapping to search and rescue. Amazon showcased its delivery drone system recently and the DJI matrice drone (shown in figure 1) was used to perform thermal inspections on windmills.

In comparison with other modes of transportation, flying vehicles have a distinct advantage of being able to control its own position in 3 dimensional space rather than be limited to a plane such as the ground or sea. Additionally, holonomic systems, such as quad-copters have the capability to quickly change their direction of motion, allowing very fast and agile flight.

Although quadcopters are being used for multiple outdoor applications, their use for autonomous indoor flight have been very limited due to many reasons. In this project we try to implement a coverage planning algorithm that can be used along with an autonomous quad-copter capable of position control. This will allow the drone to autonomously cover the entire map.

The target environment is usually split into non-intersecting regions called cells using a decomposition technique. The size and resolution of the cells may change according to the type of decomposition and the strategy applied in order to guarantee the complete coverage. Several techniques are covered in [3].

In order to execute CPP with optimal path, area/volume coverage and least resource consumption, [4] covers the following as few important requirements:

- Robot must move through all the points in the target area covering it completely.
- Robot must fill the region without overlapping paths.
- Continuous and sequential operation without any repetition of paths is required.
- Robot must avoid all obstacles.
- Simple motion trajectories (e.g., straight lines or circles) should be used (for simplicity in control).
- An “optimal” path is desired under available conditions.

Fig. 1. Amazon Air, DJI Matrice

II. History of Quadcopters

The earliest recorded use of an unmanned aerial vehicle for occurred on July 1849, when the Austrian imperial forces besieging Venice attempted to float some 200 paper hot air balloons each carrying a 24-30 pound armament. Aerial drones have been thought of even before, as seen from the numerous drawings of aerial vehicles Leonardo Da Vinci Quadcopters [9].

Quadcopters are much older than most people expect. Quadcopters were among the first vertical take-off and landing vehicles (VTOLs). Engineers developed quadcopters to solve the problems that helicopter pilots had with making vertical flights. One of the first was the de Bothezat helicopter [10] and it was an experimental quadrotor helicopter

*Equal contribution

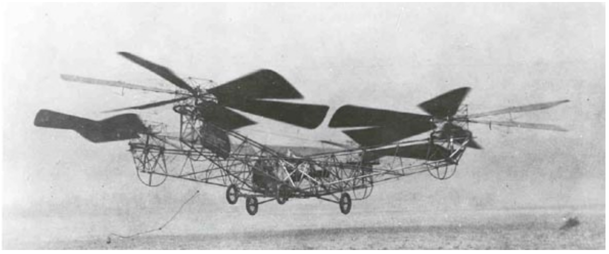

Fig. 2. De Bothezat Helicopter, 1923

built for the United States Army Air Service by George de Bothezat in the early 1920s, and was said at the time to be the first successful helicopter [11]. However,they did not become very popular due to many reasons,among which were mechanical complexity, relatively large size and weight(since they were driven by engines), and difficulties in control.

III. Previous Work

Choset [2] presents an excellent survey on CPP for mobile robots, where the author classifies the approaches either as heuristic or complete. In the heuristic approaches, the robots follow a set of simple rules defining their behavior, but such methods do not present a guarantee for coverage success. On the other hand, complete methods can provide these guarantees using the cellular decomposition of the environment, which consists of space discretization into cells to simplify the coverage in each sub-region.

Another important issue mentioned by Choset [2] is the flight time, which can be minimized, using multiple robots and reducing the number of turning maneuvers. Finally, the author highlights the available environment information. Several approaches admit previous knowledge of the robot regarding the search area (offline), while sensor-based approaches acquire such information in real-time during the coverage (online).

Choset and Pignon [5] first introduced the Boustrophedon Cellular Decomposition (BCD) family of algorithms for coverage-based path planning in an unknown environment. Their general coverage strategy extended the Seed Spreader algorithm [6] for producing back-and-forth sweeping motions through the free space. Acar et al. [8] further developed BCD-based coverage with experimental verification for a variety of control Morse functions. Mannadiar and Rekleitis [7] presented a variant of this technique which guaranteed optimal coverage for an arbitrary known environment.

The aerial robotics community has developed a number of systems that either directly achieve terrain coverage or use similar techniques to address other applications such as search and rescue. The problem of covering the area of a bounded environment while avoiding a known set of arbitrarily shaped obstacles is solved using a two-stage hierarchical solution. First, an off-line analysis of the environment decomposes the free space into a set of simple regions, called cells, and determines an Eulerian circuit through all connected cells. During the on-line stage, back-and-forth sweeping motions are generated to cover individual cells while following the Eulerian circuit. Details about this method can be found in [8].

Given that boundaries are shared between obstacles and free space, one method is to set the direction of coverage orthogonal to the dominant edge orientation for obstacle boundaries, which would arguably produce longer sweep lines as a result. An alternative strategy is to align the direction of coverage directly with the distribution of the free space, under the assumption that the length of sweep lines will be maximized along the dominant axis of the free space.

Some of the results presented in the referred paper [8] are: In figure 3 and In figure 4 the light green areas represent the area covered, and the black region is no-fly zone. Back and forth approach is used to cover the ground. We can see the traversed path as white lines and black in figure 3 and 4 respectively.

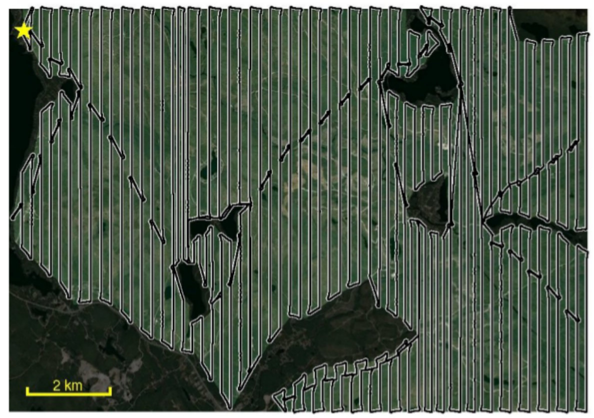

Fig. 3. Simulated coverage paths for a 13 km 10 km region at 300 m altitude

Fig. 4. Simulated coverage paths for a 1 km 0.6 km region at 150 m altitude

IV. Small Angle Control

Small angle control starts with the assumption that the quadcopters roll, pitch orientation does not vary much from the hover condition. And this assumption is valid for most practical autonomous quadcopter applications, where the drone is not expected to follow very aggressive trajectories. The small angle controller mentioned in [13] uses 2 nested feedback controllers The low level the attitude controller, which controls the roll, pitch and yaw angles of the quadcopter runs at about 1 KHz[14]. Inertial Measurement Sensors are used to obtain feedback data for the attitude controller.

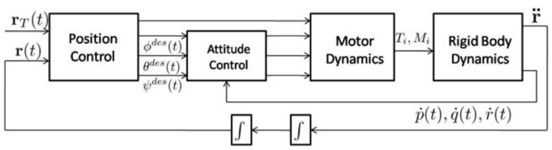

Fig. 5. Small angle control [13]

The second level nested control is the position controller whose output is fed to the attitude controller. In real-world scenarios, GPS, Camera based localization or Motion Capture Systems are used for position feedback.

In addition to these two controller, for our simulation we added a velocity controlled in between. The velocity controller helps by constraining the velocity of the quadcopter when the position setpoints are changed quickly.

A. *Control Model*

In most quadcopter control models, we assume that the motor dynamics are relatively fast, ie, the motor quickly reaches its angular speed setpoint. Including the motor increases the complexity without a lot of improvement in performance.

The state of the model is taken as the position and velocity of the center of mass and the orientation and the angular velocity.

$$X = [x, y, z, \phi, \theta, \psi, \dot{x}, \dot{y}, \dot{z}, p, q, r]^T$$

Here orientation is only locally parametrized by the Euler angles. The input is

$$u = [u1, u2, u3, u4]^T$$

where $u1$ is the total thrust from the propellers, and $u1, u2, u4$ are the moments about the body frame axes.

B. *Attitude Control*

The attitude control in a quadcopter controls the orientation of a quadcopter. To achieve a stable flight, the roll and pitch angles need to be changed in order to follow the required trajectory. Several different control methods can be used to achieve stability. Most commercial systems use LQR or PID controllers. The first controller to be used for the purpose of attitude control in the quadcopter system was the PID, later followed by the LQR controller and very recently a combination of optimal control schemes, PID controllers and other advanced controllers are being used.

1) *Proportional-Integral-Derivative Controller:*

The Proportional-Integral-Derivative or the PID controller gives the simplest solution to the various real-world problems. Both the transient and steady state response are taken are of with its three-term functionality i.e. proportional, integral and derivative. The PID is a control loop feedback system mechanism used in the majority of the Industrial control System. Since the invention of the PID controller both its usages and popularity has increased due to the advances in digital technology. The equation for the PID controller is given by

$$u(t) = K(e(t) + \frac{1}{T_i}\int_t^0 e(\tau)d\tau + T_d\dot{e}(t))$$

where $e(t) = y_{sp} - y$ is the error between the measured process variable and the reference signal, often call the set point T_i is the integral time and t_d is the derivative time. If the process variable cannot be measured, then we take the observation of the estimates of the measurable outputs.

C. *PID Attitude Control*

For any quadcopter to be in the stable and hover state the total force generated by the Quadcopter should be equal to the total forces acting on it i.e. the force must be equal to the product of its mass and gravity on the object, this is given by

$$F_i = mg$$

where F_i is the total force generated by the rotors of the Quadcopter. The Attitude control for the small angle control is discussed in this section. The attitude control is used to track trajectories in $SO(3)$ that are close to the nominal hover state where the roll and pitch angles are small. From the equation of general angular acceleration in the Motor model, if we assume that the products of Inertia are small which are Ideally the products of inertial are zero because the axes ares close to the principle axes and because of symmetry $I_{xx} \approx I_{yy}$ then the euler equation becomes

$$I_{xx}\dot{p} = u_2 - qr(I_{zz} - I_{yy})$$
$$I_{yy}\dot{q} = u_3 - pr(I_{xx} - I_{zz})$$
$$I_{zz}\dot{r} = u_4$$

we can also assume the component of the angular velocity in the z_B direction , r is small so the rightmost terms from the first two equations above, which are products involving the value of r is small compared to the other terms. We note that near the nominal hover state $\dot{\phi} \approx p, \dot{\theta} \approx q$ and $\dot{\psi} \approx r$. For these reasons we can use simple proportional derivative control laws that take the form

$$u_{2,des} = k_{p,\phi}(\phi^{des} - \phi) + k_{d,\phi}(p^{des} - p)$$
$$u_{3,des} = k_{p,\theta}(\theta^{des} - \theta) + k_{d,\theta}(q^{des} - q)$$
$$u_{4,des} = k_{p,\psi}(\psi^{des} - \psi) + k_{d,\psi}(r^{des} - r)$$

the vector of the desired rotor speeds can be found from the desired net force ($u_{1,des}$ and the moments ($u_{2,des}, u_{3,des}$ and $u_{4,des}$) by inverting we get the inverted matrix.

D. Velocity Control

Although a velocity controller is not mentioned in the small angle control scheme in [13], practical implementations of quadcopter controllers such as [15] uses a velocity controller in between the position and attitude controllers. The cascaded controller scheme used for our simulations. The equations for the velocity PID outputs are given by

$$V_{out,x} = k_{p,v_x}(v_x^{des} - v_x) + k_{d,v_x}(\dot{v_x}^{des} - \dot{v_x})$$

$$V_{out,y} = k_{p,v_y}(v_y^{des} - v_y) + k_{d,v_y}(\dot{v_y}^{des} - \dot{v_y})$$

$$V_{out,z} = k_{p,v_z}(v_z^{des} - v_z) + k_{d,\dot{v_z}}(\dot{v_z}^{des} - \dot{v_z})$$

Where v_x,v_y and v_z are the feedback values and $V_{out,x}$,$V_{out,y}$ and $V_{out,z}$ are the outputs of the velocity controller.

The velocity controller is a simple PID control for each position axis. The feedback is real-world systems are usually obtained by air flow sensors or by differentiating the position odometry values.

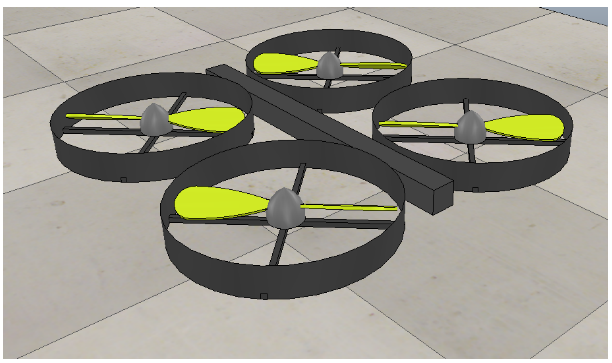

Fig. 6. Quadcopter Model

V. Method

A. Obstacle Space

For this project we are considering indoor exploration using quadcopters. Hence, our map has the environment of a house i.e. walls, tables on the ground, etc. Our main aim is to cover the floor area, hence obstacle space would include all the walls, doors, tables, etc. Since we are constraining our quad to be at an altitude and explore the house, this would not require any object below the quad to be considered as an obstacle. But, we would like to make the problem more generic and consider objects such as table, bed as obstacles as well so that in case our quad has to perform a rescue operation, it needs to know the locations of all the objects in the map in order to find an efficient path avoiding obstacles. Also, an advantage of knowing the position of the obstacles is that it can be used for emergency quadcopter landings, efficient maneuvering in 3D space.

The obstacle map we have created in V-rep is as shown in the figure 7. This is a 3D representation of our map. The problem we are solving is a 2D problem wherein the map

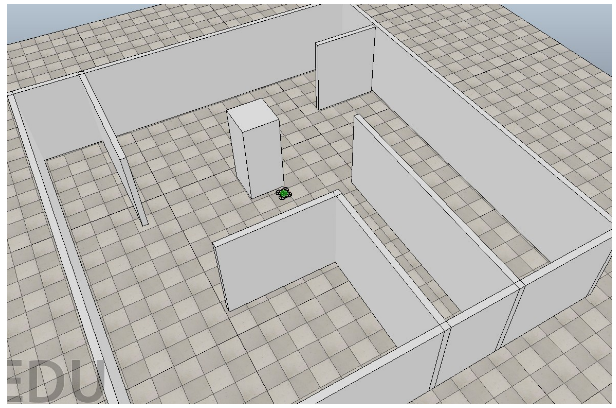

Fig. 7. Environment in V-REP

looks as shown in figure 8. Since our aim is to cover as much floor area as possible, and given that our quadcopter is restricted to an altitude, we can do planning for efficient coverage in 2D plane of the floor.

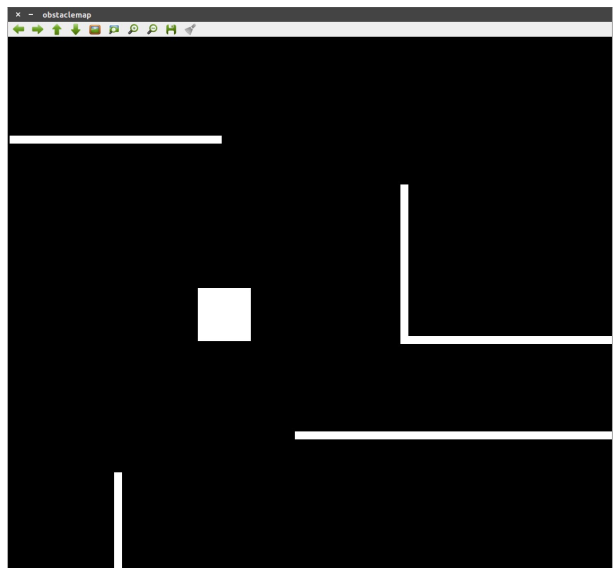

Fig. 8. 2D MAP

B. Cell Decomposition

Now that we have a 2D map of our obstacle space, we need to break it down into cells for efficient coverage path planning. In general there are two types of cell decomposition techniques used widely for Coverage Path Planning using UAVs, i.e. exact and approximate cellular decomposition. For coverage path planning, since a main criteria is to cover as much area as possible, we consider exact cell decomposition as our method for cellular decomposition. A nice illustration of the difference between area coverage possible with exact cell and approximate cell decomposition can be seen in figure 9. The grey area in the approximate cell decomposition case(right sub-image in the figure) is the area not being covered due to limitations in choosing a certain polygon type.

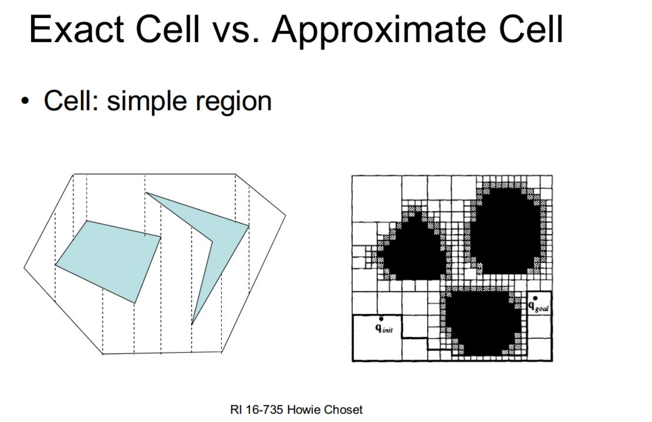

Fig. 9. Types of cells

Now, there are several types of decomposition techniques that can be applied to decompose the map into exact cells. A simple method of generating a cell is by drawing vertical/horizontal lines from the vertices of obstacles parallel to the vertical/horizontal axis. We considered and studied two types of decomposition techniques; trapezoidal and Boustrophedon cellular decomposition. Trapezoidal cell decomposition considers all vertices for generating cells. This would produce cells in greater number than necessary. Boustrophedon method takes this into consideration and generates less number of cells. The problem with trapezoidal method of cell decomposition is that it leads to more distance coverage unnecessarily due to more number of cells. This is shown in figure 10. Hence, Boustrophedon is efficient in coverage path planning as we cover same if not more area by traversing lesser distance compared to trapezoidal cell decomposition.

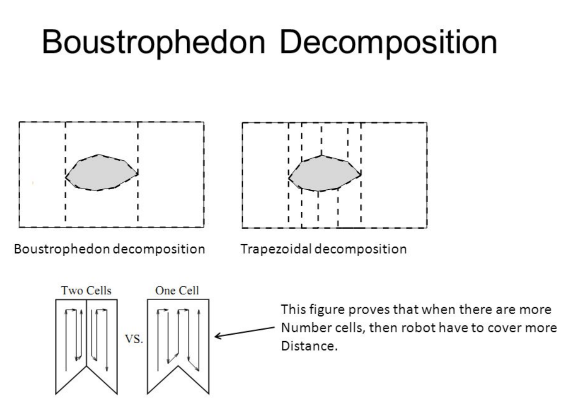

Fig. 10. Boustrophedon vs Trapezoidal Cell Decomposition

Once, the decomposition technique is applied, the resulting cells can be visualized as shown in figure 11.

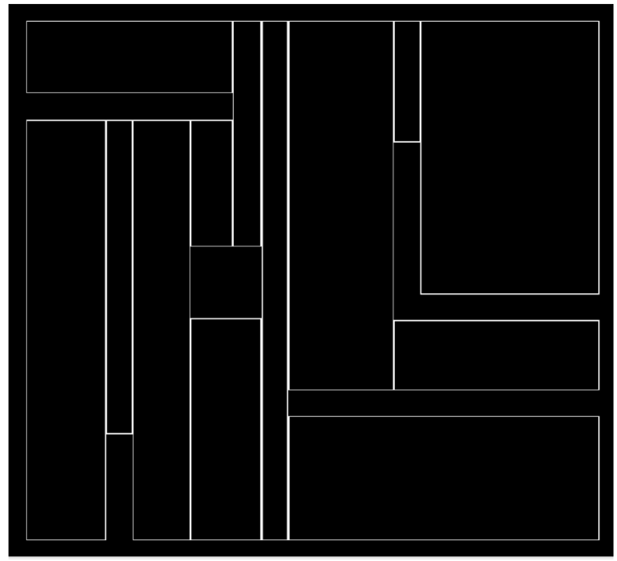

Fig. 11. Cells after Boustrophedon decomposition

C. Inside cell path planning

In this project, we are considering the resolution of the coverage area as 1cm x 1cm. This would mean that the quadcopter can cover a square of side 1cm for every step it moves. This is a valid and desirable assumption as given any area it can be reduced to this resolution. Ideally, the resolution would depend on the sensor used by the quadcopter. If a camera is being used, the field of view of the camera on the quadcopter should be considered and this can be scaled accordingly to a 1cm square grid.

Now that we have decided on the resolution and the grid size, we need to cover each grid in a cell in order to get maximum area coverage. Hence, we use back-and-forth approach within a cell to cover maximum area. This is an efficient method for coverage path planning. This saved some computation instead of utilizing complex path planning algorithms. This approach was optimized by finding the ideal sweep direction from the corner to cover the entire cell. The major and minor axes lengths were calculated from the corners of the cell, and the quad-copter's primary direction of movement was along the major axis. This helped minimize the number of turns the quad has to take and hence reduce the time to cover the area.

Also, we are extending this motion to 3 dimensions. Once the area of the cell is covered by back and forth motion, the quadcopter elevates to a higher altitude and sweeps the same area in that altitude. The number of steps and total height the quadcopter can reach is configurable. This movement in z-axis ensures volume coverage and is an important part for disaster management applications.

D. Cell to Cell path planning

After covering current cell through back and forth motion, it is important to choose the next cell to go to. There are many ways to decide which cell to go to next. One could consider the nearest cell center, nearest corner or a complex algorithm that calculates the next cell considering the entire distance that needs to be covered once it reaches that cell. For simplicity and computational efficiency, we considered nearest corner criteria. Here, we move from current position to the nearest corner to that position and traverse the cell that corner belongs to.

The path from the current position to the corner of the next cell is computed using the A* algorithm. The A* algorithm requires a start and goal node. The start node is selected as the current node. First we search for all corners within a circle of heuristic radius. If no cell corners are found within this radius, the search radius is increased by a certain amoung. This is repeated, until a cell corner is found. Then we get the all the corners detected in the radius and calculate the A* distance to all the corners from the current position. The goal node is set as the corners.The corner with the least A* distance is chosen and quad moves to the corner following the path generated by A* algorithm. Once we reach a corner of the cell, we traverse that cell using back and forth motion and mark the cell as visited. This is repeated until all the cells are marked as visited.

VI. Simulation and Results

In this section we present the simulation of the planning algorithm setup mentioned above. The simulation was done using the VREP simulation environment[1]. The python remote API was used to simplify software development. VREP contains a model of a quadcopter, which simulated the forces generated by a motor and propeller, by applying a force and corresponding torque at the locations of the motor.

The planning algorithm requires the quadcopter simulation model to move in two cases. One while executing the back and forth motion to cover the area in each cell. In the second case, the quadcopter is required to move from the corner of one cell to another. Both these algorithms would produce a sequence of position setpoints, which is simply given to the position controller in vrep sequentially.

A. A* Validation

The A* algorithm creates a path which allows the quadcopter to reach the goal without colliding into objects. This algorithm is validated if the quadcopter does not crash into the the obstacles during its movement to the goal point.

B. Coverage Planning Validation

In order to visualize that the coverage planning algorithm was successful, we marked the 3-D positions the quadcopter passed through previously with white spheres. This will give a good visualization of the positions the quadcopter has moved through before and will show whether the algorithm actually enables the quadcopter to cover the entire area.

This validation can be replicated on most computers using the open source code provided at: https://github.com/ashwinvk94/vrep_quad_exploration

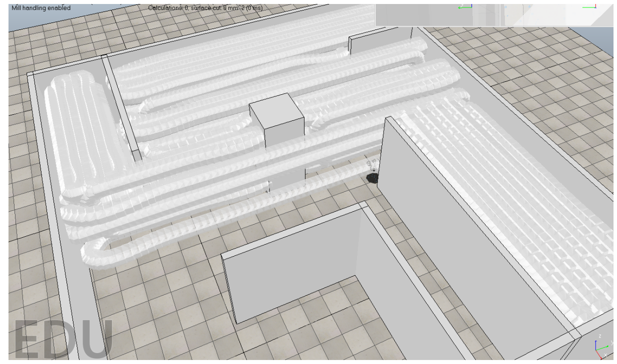

Fig. 12. Intermediate Coverage

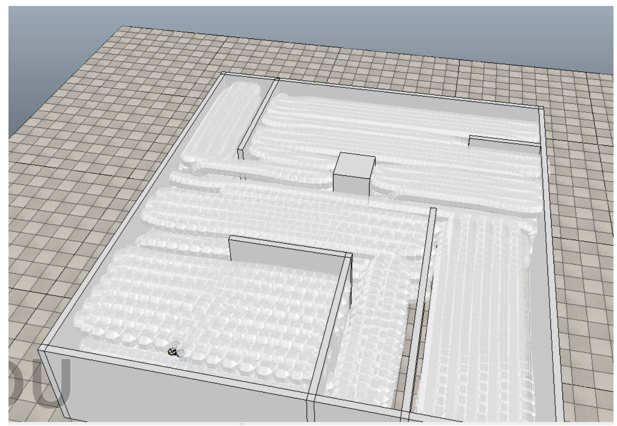

Fig. 13. Final Coverage

VII. Conclusion

We successfully created an algorithm to enable the quadcopter to cover the 3-D space in the VREP environment. In order to verify successful coverage, we calculated the area filled by the white spheres, corresponding to the previous locations of the quadcopter. We found that approximately 95 percent of the area was covered(not including clearance area). The final coverage planning video can be found here.

References

[1] E. Rohmer, S. P. N. Singh, M. Freese, "V-REP: a Versatile and Scalable Robot Simulation Framework," IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 2013

[2] Choset, H. Coverage for roboticsA survey of recent results. Ann. Math. Artif. Intell. 2001; pp. 113126.

[3] Taua M. Cabreira, Lisane B. Brisolara, Paulo R. Ferreira - Survey of Coverage Path Planning with Unmanned Aerial Vehicles. 2019

[4] Enric Galceran and Marc Carreras-A Survey on Coverage Path Planning for Robotics. 2013

[5] H. Choset and P. Pignon, Coverage Path Planning: The Boustrophedon Cellular Decomposition, in In Proc. of the Int. Conf. on Field and Service Robotics, Canberra, Australia, 1997.

[6] V. J. Lumelsky, S. Mukhopadhyay, and K. Sun, Dynamic path planning in sensor-based terrain acquisition, IEEE Trans. on Robotics and Automation, vol. 6, no. 4, pp. 462472, Aug. 1990.

[7] R. Mannadiar and I. Rekleitis, Optimal Coverage of a Known Arbitrary Environment, in Proc. of IEEE Int. Conf. on Robotics and Automation, Anchorage, AK, May 2010, pp. 55255530.

[8] Anqi Xu, Chatavut Viriyasuthee, and Ioannis Rekleitis "Optimal Complete Terrain Coverage using an Unmanned Aerial Vehicle"

[9] https://airandspace.si.edu/stories/editorial/leonardo-da-vinci-and-flight

[10] Young, Warren R. (1982). The Helicopters. The Epic of Flight. Chicago: Time-Life Books. p. 28.
[11] J.G. Leishman, The Breguet-Richet Quad-Rotor Helicopter of 1907.
[12] Howie Choset. Lecture on Cellular decomposition
[13] Mellinger, Daniel Warren, ”Trajectory Generation and Control for Quadrotors” (2012). Publicly Accessible Penn Dissertations. 547.
[14] D. Gurdan, J. Stumpf, M. Achtelik, K. Doth, G. Hirzinger, and D. Rus. Energy- efficient autonomous four-rotor flying robot controlled at 1 khz. In Proc. of the IEEE Int. Conf. on Robotics and Automation, Roma, Italy, April 2007
[15] Meier, Lorenz Tanskanen, Petri Fraundorfer, Friedrich Pollefeys, Marc. (2012). The PIXHAWK open-source computer vision framework for MAVs. ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. XXXVIII-1/C22. 13-18. 10.5194/isprsarchives-XXXVIII-1-C22-13-2011.