USE OF UNMANNED AERIAL VEHICLE (UAV) PLATFORM FOR 3D MAPPING AND NAVIGATION OF SUBTERRANEAN LANDSCAPE USING 3D LIDAR SENSOR

Malith Kuruppu, Yuthika Sagarage, J. K. Jayawardena
Communication Division
Arthur. C. Clarke Institute for Modern Technologies
Katubedda, Moratuwa, Sri Lanka
kavindra@accmt.ac.lk

***Abstract*— This paper presents the formalization of the hardware and software implementation for an Unmanned Aerial Vehicle which will be used for 3D mapping and real-time 2D obstacle avoidance using a 3D Light Detection and Ranging sensor. The proposed implementation can be used to map and navigate in areas with low visibility, limited or no Global Navigation Satellite System coverage and without any prior information about the site such as a in a subterranean landscape. The navigation is based on a Vector Polar Histogram Plus, but the "construction of cost function" is modified to navigate through unknown environments. This also means that the system is fully autonomous. The 3D mapping is based on "Velodyne SLAM", which simultaneously localizes and maps its environment in 3D.**

Keywords— UAV, LIDAR, Subterranean Navigation, 3D Mapping

I. INTRODUCTION

Subterranean locations such as mines are extensively being blasted to explore new mining ores. During this process, the mine needs to be repeatable and systematically measured, which traditionally is done by taking point measurement manually using a total station in a dangerous environment. Very often ores are blasted to create vertical space between two tunnels that result in cavities known as "stopes". It is very dangerous and nearly impossible to map these underground cliffs.

One way of mitigating this problem is by using UAVs, which can potentially eliminate the need for human intervention. UAVs are generally is easy to transport and set up, compared with other underground mapping systems. In the recent international scenario, UAVs are finding its way to every single industry some way or another, be it agriculture, survey, forestry, defence surveillance, which is overground. But it is a completely different story in the underground landscape such as in pipelines, tunnels and mines where there is no Global Positioning System (GPS) and little or no light. In this scenario, off-the-shelf UAVs are useless with limited capabilities.

The main problem to be addressed for UAVs to function effectively underground is by having a system that can boost the spatial awareness of the subterranean landscape and use acquired spatial information to navigate autonomously in the terrain. One way to solve the problem of mapping and navigation in a subterranean landscape is by using a Light Detection and Ranging (LIDAR) sensor. LIDAR is a remote sensing method by which a beam of light in the form of a pulsed laser is used to measure the distance from the light source. The data collected by the LIDAR sensor in the form of a point cloud is fed into a powerful onboard computer to perform Simultaneous Localization and Mapping (SLAM) algorithm, which is an odometry methodology used by autonomous systems to construct a map of an unknown environment while simultaneously keeping track of the system's location relative to the unknown environment.

This enables the UAV to navigate around a subterranean environment with omnidirectional collision avoidance in a GPS denied location. This gives the UAV a high level of autonomy where the UAV can operate well beyond line-of-sight without prior information on the terrain, human intervention and communication/telemetry link. Additionally, production of a SLAM based 3D map of the environment provides key stakeholder with valuable measurement data of the environment and gives explorers a visual aid for further investigation.

This method of path planning and surveying of underground mine using LIDAR technology on a UAV platform will drastically increase the quality of maps acquired while reducing the risk of human harm and time taken to obtain such results. When such system is completed implemented with repeatable and accurate results, this system can be deployed in a fraction of time to environments other than underground with no or little access to humans, dangerous for human intervention and GPS coverage such as inside buildings during a disaster situation, construction sites and so on.

II. HARDWARE OVERVIEW

The UAV platform proposed for the implementation is a carbon fibre quadcopter with a radius of approximately 650mm, with an open-source autopilot, Pix4. The UAV is proposed to be equipped with a 360° 3D LIDAR, Velodyne Lidar PUCK VLP16 [1] as both a 3D point cloud generating device for mapping and a ranging measuring sensor for navigation.

The working principle behind a LIDAR is based on Time of Flight (TOF) measurement [2]. It emits a pulsed laser beam and time taken for the beam to be emitted by the sensor, reflected by the surface and finally received by the sensor [2].

III. IMPLEMENTATION OF NAVIGATION

Since the UAV needs to traverse through subterranean landscapes which do not have any GPS or light, a LIDAR based navigation is proposed. The proposed LIDAR also serves as a tool to obtain a 3D point cloud map of the landscape.

For the proposed navigation it is assumed that there is no sudden change in the landscape around the vertical axis. Therefore, the problem of navigation can be reduced to 2D obstacle avoidance. Thereby, only a layer of LIDAR data is obtained and processed.

The implementation of navigation takes in the following sequence:

1. Takeoff from Home
2. Get Current LIDAR data
3. If the battery life to return from the current location is takeoff location is less than available battery life then return Home and land. Or else continue.
4. Implementation of Local Obstacle Avoidance using VPH+
5. Find the next best candidate direction and location
6. If more than one candidate direction, store the current location, next direction and other possible direction into the flight computer. We will call this location as Branching Point (BP)
7. Continue to the next location and repeat 3-7
8. If there is a dead-end, come back to the last BP and select other possible directions and repeat 3-7. Until the first BP
9. Return to the Home and land

A. Local Obstacle Detection and Categorization

The algorithm proposed for local obstacle avoidance is part of Enhanced Vector Polar Histogram (VPH+) which is obstacle avoidance model for which ideal for data obtained via a Light-based ranging sensor such as a LIDAR.

In every path planning cycle, the algorithm employs only four steps from the original paper [3] : (a) modification of original data, (b) grouping of obstacle into clusters, (c) construction of a threshold function, (d) construction of a symbol-function.

The algorithm for the 4 steps is given below:

(a) Modification of original data:

The raw data from the LIDAR is modified to correct for the maximum distance the robot can reach in a given direction.

1. Get LIDAR distance data for directions of i (i=0,1,2,…,360) as in Fig. 1.
2. Calculate s_{ij}

$$s_{ij} = d_j \sin y_{ij} \quad (1)$$

3. Assuming that the robot's radius is R. If R < s_{ij} OR R≥ s_{ij} AND d_i < d_j $\cos y_{ij}$, modified distance of the current obstacle point is d_i. Otherwise, the modified distance of the current obstacle point is d_j $\cos y_{ij}$.
4. The farthest distance that the robot can travel, Di in the direction i is: modified distance of current obstacle point from 3 subtracted by the robot's radius.

(b) Grouping of obstacle into clusters:

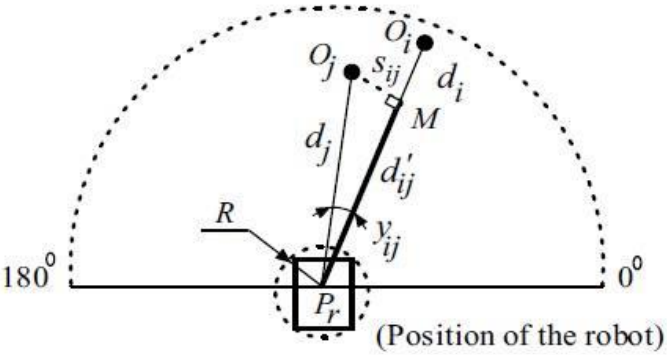

Fig. 1 Illustration of Robot and two obstacles

In the actual environment, an obstacle usually includes a set of obstacle points. In our study, we regard these obstacle points that have a close distance between each other as a block. In this way, the environment and the computation can be simplified. The principle to group the obstacles into blocks is as follows:

1. Get LIDAR distance data for directions of i (i=0,1,2,…,360)
2. Find the distance between two adjacent obstacle points (d_{ij}) ..

$$d_{ij} = \sqrt{{d_i}^2 + {d_j}^2 - 2d_i d_j \cos y_{ij}} \quad (2)$$

3. If d_{ij}<d(threshold) then the two-point belong to the same block, which means that the robot cannot pass through the gap between the two-obstacle point.
4. If d_{ij} ≥d(threshold) then the two-point are regarded as of two different blocks which means that the robot can consider the gap between the two-point as a potential passage for the robot to go through.

(c) Construction of a symbol function:

The role of the symbol function is to reveal the geometrical relationship of each grouped obstacle block and to mark every obstacle point with binary data.

In the VPH+ method, obstacles are grouped into blocks which are classified as concave block or non-concave block. The concave block means that it is closer to the robot, therefore, the corresponding directions which the concave blocks lie in should be excluded from the candidate

directions for the robot. A block is concave or not depends on its adjacent blocks.

1. Get modified distance data from the edges of the blocks.
2. If the modified distance of edges of the current block is less than previous and the next block then the block is concave. Therefore, for every obstacle point in a concave block, i the symbol function, B(i) value should be 0. Otherwise, the B(i) should be 1.

(d) Construction of a threshold function:

From the view of kinematics, the main purpose of the threshold function is to make sure that there is no crash with obstacles when the robot is moving at a certain speed by setting a safe distance.

1. If $D_i \geq d_{safe}$ (calculation of d_{safe} .), then threshold function, H(i) is equal to 1 or otherwise it is equal to 0.

$$d_{safe} = K\left(\frac{V}{v_{max}} R_{min} + vT\right) \tag{3}$$

Where: K is a safety coefficient (K=1.2 ~ 1.5);
v_{max} is the maximum speed of the robot;
v is the speed of the robot(v)
R_{min} is the minimum steering radius at v_{max};
T is the planning cycle of the robot.

With:

v_{next} is the speed of the robot(v) for the next planning cycle:

$$v_{next} = \begin{cases} v_{max}, & \sqrt{2aD_m} > v_{max} \\ \sqrt{2aD_m}, & others \end{cases} \tag{4}$$

Where: D_m is the modified distance in the direction of travel;
a is the maximum acceleration of the robot.

B. Choosing the next moving point

The original paper was based on goal-oriented navigation and the aim to taking the heading deviation and speed into account, to formulate a time-oriented cost function for the robot to get to the goal position in a shorter time. But in our proposal, there is no predefined goal, but the robot needs to map the terrain by exploring it freely. At the beginning of the exploration, there are no branches, k=0.

For exploration navigation using VPH+ information, the following steps are taken:

1. If B(i)H(i) is one in the direction of the current heading, then continue the moving in the current heading.
2. Otherwise, if $B(i)H(i)D(i) \geq 0.5D_{max}$, then cost function, C(i) is constructed with a value 1 or else is 0 for every point i.
3. If only one point with C(i) with 1 then go in the direction of i.
4. If multiple points with C(i) equals 1, then at increase k by one and record the number of sub-branches at the kth branch, SB_k.
5. Repeat 1-4 until dead-end

C. Deadends and return home

If the robot encounters a dead-end, that the robot cannot traverse any further in its path then the following steps will be executed by the robot:

1. If the C(i) is 0 for all direction point, i, then return the robot to the previous branch point.
2. If in the kth branch, there are more sub-branches unexplored then move to the next sub-branch, else move to the previous branch until the robot is in the home.

IV. IMPLEMENTATION OF 3D SIMULTANEOUS LOCALISATION AND MAPPING

In Parallel with the obstacle avoidance algorithm, the 3D SLAM algorithm is also implemented. Fundamentally, SLAM is a mathematical model that enables a robotic system traverse through an unknown environment without any prior topological knowledge of the location or any artificial infrastructure [4]. SLAM can be broken down into two segments: Mapping which constructs a map of an unknown environment and Localization which is calculating the position of the UAV relative to the unknown environment [5].

SLAM helps build a global 3D map of the environment, which is used for localization which ultimately enables the UAV to “remember” branch and sub-branch location relative to its current location. An additional advantage of the SLAM is that the 3D map generated can be further post-processed to give a high-quality render of a point cloud of the topology.

The proposes SLAM implementation is based on Velodyne SLAM [6]. In our work, a Velodyne Lidar PUCK VLP-16 laser scanner is used which is mounted on top of the UAV. An overview of the Velodyne SLAM method is explained herein.

A. Scan Acquisition and Pre-Processing

The data acquired from a single scan (scan index k), which is an array of range measurement into range image.

1. The 2d array of range measurements R: (u,v) -> r. For simplicity, the pixel coordinates R(u,v) are substituted with a subscript function by index R_i. The connections between the current pixel to its four neighbours are denoted by the following indices [6]. Some of the relations are illustrated in Fig. 2.

$$i_1 := (u+1, v)\ ;\ i_3 := (u-1, v)\ ;\ i_5 := i_1$$
$$i_2 := (u, v-1)\ ;\ i_4 := (u, v+1)\ ;\ i_{1_1} := ((u+1)^{+1}, v)$$

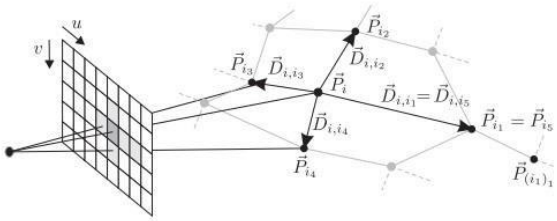

Fig. 2 Range image as implicit graph on 3D coordinate [6]

2. The point coordinates, $\vec{P_i}$(relative to the laser scanner) are obtained from the range measurements using the physical set-up. The distance vector, $\overrightarrow{D_{i,j}} = (\vec{P_i} - \vec{P_j})$ is calculated [6].
3. A linkage value, $L_{i,j}$ is a measure of how likely a pixel is in the same plane as its neighbouring pixels and is used for a weight calculation on pixel connection. For example, the linkage value of a pixel to its right neighbour is calculated as [6]:

$$L_{i,j} = \min\left(sigm\left(\frac{(R_i - R_{i_1}) - (R_{i_3} - R_i)}{(R_{i_3} - R_i)}\right), sigm\left(\frac{(R_i - R_{i_1}) - (R_{i_1} - R_{i_{1_1}})}{(R_{i_1} - R_{i_{1_1}})}\right)\right) \quad (5)$$

The sigmoid function is used as a threshold function:

$$sigm(x) = 0.5 - \frac{0.5(x - \theta_1)\theta_2}{\sqrt{1 + (x - \theta_1)^2 \theta^2{}_2}} \quad (6)$$

Where, θ_1: Effective threshold
θ_2: Constant scale parameter

4. Local surface plain by its normal vector, $\overrightarrow{N_i}$ is estimated at each measurement. Then a moving average filter is then applied to the field of the surface normal in order to reduce noise [6]:

$$\overline{N_i} = \frac{\sum_{j=1}^{4} \overrightarrow{N_i}'}{\left\| \sum_{j=1}^{4} \overrightarrow{N_{i_j}}' \right\|} \quad (7)$$

Where

$$\overrightarrow{N_i}' = \sum_{j=1}^{4} L_{i,i_j} L_{i,i_{j+1}} \left(\overrightarrow{D_{i,i_j}} \times \overrightarrow{D_{i,i_{j+1}}}\right) \quad (8)$$

5. A confidence value, $C_{i,j}$ is estimated for each normal value. For a given connection from i to j, the angle of the distance vector to the plain defines the probability that the plain assumption holds for this connection [6]:

$$C_{i,j} = exp\left\{-\theta_3 \arcsin\left|\frac{\overrightarrow{D_{i,j}} . \overrightarrow{N_i}}{\|\overrightarrow{D_{i,j}}\|}\right|\right\} \quad (9)$$

Where, θ_3: Decay of the probability

6. The confidence value is limited by the maximum linkage product from the normal calculation [6]:

$$C'_i = \min(L^{max}{}_i, max\ (C_{i,i_1}, C_{i,i_3}, C_{i,i_2}, C_{i,i_4})) \quad (10)$$

Where

$$L^{max}{}_i = max_{j=1}^{4} L_{i,i_j} L_{i,i_{j+1}} \quad (11)$$

B. Localisation

The goal of this step is to estimate the pose, $\vec{v}(t)$ of the UAV relative to a global coordinate frame, which is the position and orientation of the UAV. Since, the pose of the UAV at the end of the current scan, $\vec{v}^E{}_k$ is equal to the pose of the UAV at the start of the next scan, $\vec{v}^S{}_{k+1}$, thereby it is sufficient for a given scan k only to estimate $\vec{v}^E{}_k$ [6].

1. The current vehicle position is estimated by calculating for a, $\vec{v}^E{}_k$ that gives the minimum energy[7].

$$E(v) = \sum_{i \in scan} (\vec{n}^{\mathrm{T}}{}_{NN(i)} (\mathrm{T}(\vec{P_i}, v) - \vec{p}_{NN(i)}))^2 \quad (12)$$

And

$$\vec{v}^E{}_k = argmin_v E(v) \quad (13)$$

2. Then an Iterative Closest Points (ICP) algorithm is used as described in [8].

C. Mapping

The mapping process is executed by combining the collection of surface in a world reference frame$\{s_i = (\vec{p_i}, \vec{n_i}, \vec{c_i}\}$, which were calculated in section IV-B. The map is stored in a 3D grid with a grid resolution of g, where each cell contains a surface according to its 3D position, $\vec{p_i}$ [6].

1. Each measured surface, S_i is updated according to its neighbours on the map. A weighted point-to-plain-energy is defined similarly to the ICP-energy in [6]

$$E_{S_i}(a) = \sum_{j \in kNN(S_i)} w_{ij} (\vec{n}^{\mathrm{T}}{}_j (\vec{P}'_i(a) - \vec{p}_j))^2 \quad (14)$$

$$w_{ij} = C_i c_j \overrightarrow{N_i^T}\ \vec{n_j} \quad (15)$$

Where kNN: k map-surfaces in a specified neighbourhood
wij: weights according to normal confidence and similar normal direction a: adjustment value is determined by

$$\hat{a_i} = argmin_a E_{S_i}(a) \quad (16)$$

2. The 3D point coordinate, P'_i is moved to a new point along the normal vector, N'_i to $\vec{P'_i}(a)$ where there is high normal confidence [6].

$$\vec{P'_i}(a) = \vec{P'_i} + a\,.\,\vec{N'_i} \quad (17)$$

3. Each measurement is added to the map. If the corresponding cell is non-empty, its surface, s_i is replaced by the current measurement S_i in case hold[6].

$$\frac{r_i - R_i}{r_i} + (C_i - c_i) > \theta_4 \tag{18}$$

V. PROPOSED EXPERIMENT

The navigation and SLAM algorithm will be tested in Gazebo [9], an open-source robotics simulator with a robust physics engine. In the navigation experiment, the goal was to verify the ability of the UAV to successfully explore a GPS and light denied indoor environment with multiple branching points using only a 3D LIDAR as an external sensor without any collision to an obstacle.

After satisfactory results, the SLAM algorithm will be tested by flying the UAV in manual mode in a real-life flight experiment at an indoor environment. In the mapping experiment, the goal was to compare the quantitative difference between the 3D maps generated by the 3D LIDAR against the ground truth. The UAV will be then tested in a controlled laboratory environment with both navigation and SLAM algorithms implemented, which means the UAV will be tested in a fully autonomous mode. Finally, after robust and repeatable results the UAV will be tested in subterranean mine with ground truth data pre-measured using an accurate survey-grade laser scanner.

VI. CONCLUSION

The papers present a 2D obstacle avoidance and 3D mapping using only a 3D LIDAR mounted on a multicopter. The proposed UAV is a quadcopter with carbon fibre structure which is powered by a Pixhawk flight controller and Nvidia Jetson TX2, an onboard GPU accelerated mobile computer platform. The sensor proposed is a Velodyne Lidar PUCK VLP-16. The proposed navigation implementation uses VPH+ with few modifications to the mathematical model enabling it to explore unknown environments. The proposed mapping uses a SLAM algorithm known as Velodyne SLAM.

VII. ACKNOWLEDGEMENT

The project is funded as internal research at Arthur C. Clarke Institute for Modern Technologies, through Science Research and Technology Ministry of Sri Lanka.

REFERENCES

[1] Velodyne Lidar, "PUCK," Velodyne Lidar, 2019. [Online]. Available: https://velodynelidar.com/vlp16.html. [Accessed 03 June 2019].

[2] B. Chen, Z. Cai, Z. Xiao, J. Yu and L. Liu, "Real-time detection of dynamic obstacle using a laser radar," in *The 9th International Conference for Young Computer Scientists*, Hunan, 2008.

[3] J. Gong, Y. Duan, Y. Man and G. Xiong, "VPH+: An Enhanced Vector Polar Histogram Method for Mobile Robot Obstacle Avoidance," in *International Conference on Mechatronics and Automation*, Harbin, 2007.

[4] M. W. M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte and M. Csorba, "A solution to the simultaneous localization and map building (SLAM) problem," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 3, pp. 229-241, 2001.

[5] G. Tuna, K. Gulez, V. C. Gungor and T. V. Mumcu, "Evaluations of different simultaneous localization and mapping (SLAM) algorithms," in *IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society*, Montreal, 2012.

[6] F. Moosmann and C. Stiller, "Velodyne SLAM," in *IEEE Intelligent Vehicles Symposium (IV)*, Baden-Baden, 2011.

[7] D. Holz and S. Behnke, "Sancta Simplicitas – On the efficiency and achievable results of SLAM using ICPBased Incremental Registration," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Anchorage, 2010.

[8] Y. Chen and G. Medioni, "Object modeling by registration of multiple," in *IEEE International Conference on Robotics and Automation (ICRA)*, Sacramento, 1991.

[9] " Why Gazebo?," Open Source Robotics Foundation, 2014. [Online]. Available: http://gazebosim.org/. [Accessed 13 June 2019].