Dynamic real-time indoor environment mapping for Unmanned Autonomous Vehicle navigation

Christo Aluckal
Dept. of Computer Engineering
Fr. Conceicao Rodrigues College of Engineering
Mumbai, Maharashtra, India
christo12aluckal@gmail.com

Yash Turkar
Dept. of Computer Engineering
Fr. Conceicao Rodrigues College of Engineering
Mumbai, Maharashtra, India
yashturkar@gmail.com

Yashom Dighe
Dept. of Computer Engineering
Fr. Conceicao Rodrigues College of Engineering
Mumbai, Maharashtra, India
yashom7@gmail.com

Sumedh Deshpande
Dept. of Computer Engineering
Fr. Conceicao Rodrigues College of Engineering
Mumbai, Maharashtra, India
sumedh.g.deshpande@gmail.com

Prof. B.K Mohan
CSRE
Indian Institute of Technology Bombay
Mumbai, Maharashtra, India
bkmohan@csre.iitb.ac.in

Dr. Yogesh Agarwadkar
InfiCorridor Solutions Pvt. Ltd.
Mumbai, Maharashtra, India
agarwadkar@inficorridor.in

Prof. Sunil Surve
Dept. of Computer Engineering
Fr. Conceicao Rodrigues College of Engineering
Mumbai, Maharashtra, India
surve@fragnel.edu.in

Prof. Brijmohan Daga
Dept. of Computer Engineering
Fr. Conceicao Rodrigues College of Engineering
Mumbai, Maharashtra, India
daga@fragnel.edu.in

***Abstract*— Unmanned aerial and ground vehicles today excel at autonomous navigation due to their reliance on GNSS sensors. GNSS sensors are an unviable option for navigation in indoor environments due to lack of direct line-of-sight with satellites and environmental awareness. This paper introduces a robust method for unmanned autonomous vehicles (UAVs and UGVs) that generates a knowledge base in the form of a pointset about an indoor environment using geometric primitives and range imaging sensors. The method uses depth data from range imaging sensors along with pose data from visual odometry sensors to generate a map of the environment while exploring in order to build a knowledge base. This knowledge base can be used to navigate in the mapped environment while avoiding obstacles and finding optimal route from start to goal. This paper discusses the efficacy of the proposed method, it also demonstrates results generated by implementing the method in a controlled indoor environment.**

Keywords*— *UAV Navigation, Indoor Navigation, Depth imaging, V-SLAM Navigation, Environment Mapping, 2D Mapping, Robotics.

I. INTRODUCTION

Unmanned Autonomous Vehicles or mobile robots, also known as UAVs (Unmanned Aerial Vehicles) and UGVs (Unmanned Ground Vehicles) are highly effective for a multitude of applications. They are extremely customizable and can be designed to carry various payloads such as cameras, LIDAR sensors and a variety of other sensors (optical, thermal, multispectral and hyper-spectral imaging sensors). This results in these robots having applications in various fields like surveying, agriculture, archaeology, emergency response and disaster management, mapping and reconstruction, civil and military reconnaissance to name a few. The ability to carry payloads and cover large distances makes them ideal for cargo delivery in remote places which are difficult to reach and time consuming to access by roads e.g. to deliver medicines and vaccines to doctors in remote locations.

Micro UAVs in particular have gained popularity in the past decade due to their maneuverability, mechanical simplicity, small size and affordability. These micro UAVs are especially popular in swarm robotics which involves using multiple such UAVs and having them synchronize to perform a coordinated task faster than single UAV.

Due to their reliance on Global Navigation Satellite System (GNSS) sensors (GPS, GLONASS, etc.) for localisation, these robots perform excellently in outdoor environments. GNSS satellites broadcast their location and GNSS sensors use these broadcasted signals to triangulate their own location w.r.t to the GNSS satellites. Hence, direct line-of-sight is required for GNSS to operate reliably. Satellite imaging also provides a very accurate and detailed representation of the earth thus providing the data required (road maps and locations) for navigation which is in turn used by these robots to localise themselves in their environment/s.

Localisation in indoor environments is much more challenging because of high complexity of obstacles and clutter. Lack of line-of-sight prevents GNSS to provide reliable information for the vehicle to localise itself, GNSS also fails to provide any details of possible obstacles in the environment. Although previous approaches to develop a robust indoor navigation system such as ultrasonic beacons (SEKI et al. 1998), WPS (Wi-Fi Position System) (PARK 2014), Bluetooth based approaches (SATAN 2018), etc. exist, all of them require specialised hardware to be setup in the indoor environment and hence are not dynamic in nature and cannot be used in unexplored areas. Moreover, they need to communicate with a base station which makes them unusable in remote inaccessible areas, areas such as mines or caves.

Micro UAVs and UGVs have potential in indoor applications such as but not limited to survey operations in dangerous environments and structural analysis. To achieve localisation in indoor environments a reliable and robust navigation technique is required, which does not rely on GNSS or any other beacon based system (WPS, Ultrasonic beacons); a system that is self-sufficient, dynamic and robust enough to be deployed in real-life scenarios with minimal setup and no prior knowledge of the indoor environment.

The proposed method uses depth data from range imaging sensors and pose data from visual odometry sensors (V-SLAM sensor) and generates pointset which can be easily referenced later on. The method takes into account possible obstacles in the environment and is capable of operating in complete isolation , i.e. without communication with a base station. The method proposed is lightweight and simple, making it ideal to be run on-board the vehicle using minimal computing power. This makes the system completely independent and highly dynamic as no setup is required prior to operation.

II. Methodology

A. Data Acquisition

The range imaging sensor outputs data in a pipeline of frames. Each frame provides an R, G, B and depth value per pixel. Only one row in each depth frame is considered. Isolating and extracting only one row results in construction of a 2D-map at that altitude. This isolation is performed because the robot operates at a fixed altitude, making obstacles above and below the chosen altitude irrelevant. This is referred to as *plane slicing* and is demonstrated in figure 1. The choice of row depends on the application and can be varied.

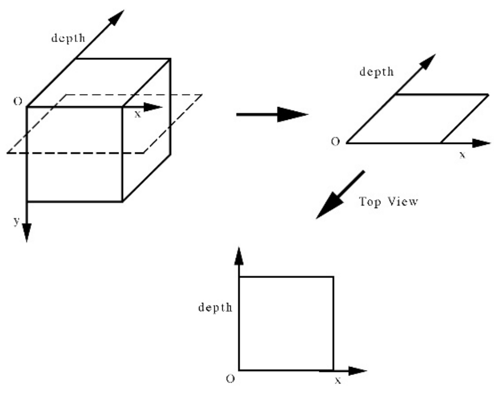

Figure 1 - Plane slicing

B. Pre-Processing

The range imaging sensor provides depth values normal to the plane of the sensor. These values need to be converted to radial distance from the center of the sensor in order to convert spatial data to a spherical coordinate system for generating a visual representation. This is achieved. using basic Euclidean geometry and trigonometry

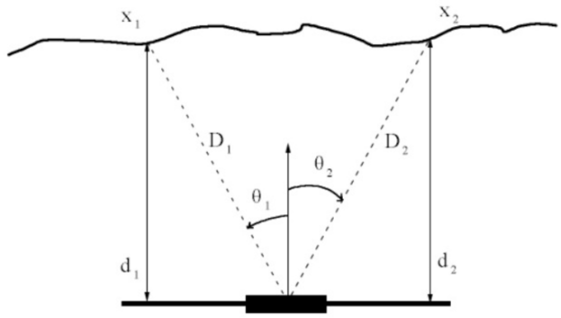

Figure 2 - Conversion to spherical coordinate system

The sensor provides the normal distance d (refer figure 2). By trigonometry, the radial distance D is the normal distance d divided by the cosine of θ.

$$D = d/\cos\theta \tag{1}$$

The sensor has a horizontal resolution of n pixels. All of these pixels have an associated distance d_1, d_2, d_3…. d_n and angles θ_1, θ_2, θ_3….θ_n made with the vertical axis, which are used to calculate their respective radial distances D_1, D_2, D_3…. Dn. Thus, for every frame in the pipeline, the radial distances are calculated for all pixels in the chosen row.

θ, for every pixel is calculated using the horizontal field of view h_{fov} of the sensor. The h_{fov}, which is a known value, is the angle between extreme pixels. The vertical axis is assumed to be aligned with center pixel and bisects h_{fov}. Counter-clockwise angles from the vertical axis to the left extreme are positive whereas clockwise angles from the vertical axis to the right extreme are negative. This makes the angle of the leftmost pixel to be $\left(\frac{h_{fov}}{2}\right)$ and the rightmost pixel to be $-\left(\frac{h_{fov}}{2}\right)$ and that of the center pixel to be 0. Thus, by elementary mathematics the generalized formula of θ_k for some k^{th} pixel is given by

$$\theta_k = \left(\frac{h_{fov}}{2}\right) * \left(\frac{n}{2} - k\right)/\frac{n}{2} \tag{2}$$

where, n is the frame width in pixels.

C. Proposed Method

The initial location of the robot is taken as the origin for the map and the pose estimation sensor (visual odometry sensor). The pose estimation sensor returns the current position of robot relative to this origin (initial location).

Once a frame has been retrieved, two operations are performed

1. Thresholding and Selective omission
2. Offsetting

Thresholding is the process of checking if the robot has translated or rotated in the 2D space from its previous point of mapping and if the data in the current frame has been already mapped. The motion of the robot is verified using the pose estimation sensor which is expected to have an inertial measurement unit (IMU) that reports its attitude and position in the 3D-space relative to the starting point. The existence of knowledge is verified by referring to the existing pointset.
If thresholding determines that the robot's position or attitude has changed by a significant margin and that the knowledge of pixels in current frame does not exist in the data structure, spherical coordinates, as explained above, are calculated for those pixels. Here the thresholding function also *selectively omits* the pixels that have already been mapped in the case of an overlap as shown in figures 3 and 4. Pixels between x_2 and x_3 are not processed again even though they are in the field of view, as they were already considered and added to the map previously.

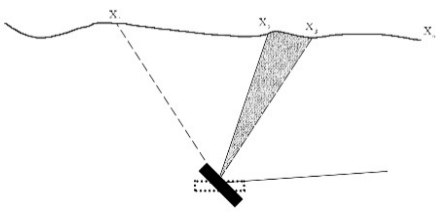

Figure 3 - Selective omission for rotation

After calculating the spherical coordinates of the pixels, they need to be *offset* because the range imaging sensor always gives depth from its current position. Therefore, if the robot translates or rotates, the radial distance or the angle with the vertical axis or both change relative to the starting point.
This needs to be compensated for and is done using the IMU and position data from the pose estimation sensor using standard homogenous translation and rotation matrices.
E.g. We create a translation matrix

$$\begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ posx & posy & 1 \end{matrix}$$

After multiplication with current position matrix, we get the final coordinates of the points relative to the starting point.
Once the final coordinates are calculated, they are added as per the required format to the custom data structure and are plotted on the map using coordinate geometry for visual representation and verification.

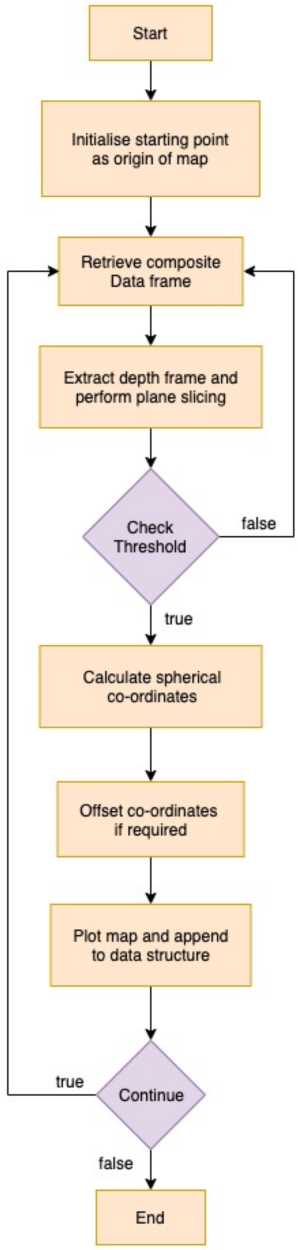

Figure 4 – Flowchart

D. *Implementation Details*

The method proposed in this paper was implemented using Intel RealSense D435 depth camera as the range imaging sensor and Intel RealSense T265 tracking camera as the visual odometry sensor for pose estimation. The Intel RealSense SDK was used to interface with the sensors and the method was implemented using Python.

The camera has a horizontal field of view of 86 degrees and is set to a resolution of 640x480 pixels at 30 frames per second. This results in the value of n being equal to 640. Substituting these values in equation (2) makes the formula for θ_k as

$$\theta_k = 43 * (320 - k)/320$$

Thresholding –

- At initialization, the sensor's current position and angles are stored in a list which is then appended to a list called $prevList$.
- When the threshold function is called, the newer position and angle values are compared with the last appended value of $prevList$.
- If the differences in horizontal/vertical movement is greater than 3.5cm or angle moved is greater than 5 degrees, the threshold function returns true which results in the newer position and angle values to be appended to $prevList$ which will be used for further iterations.
- If criteria are not met, then the function returns False which doesn't update $prevList$.

Selective omission –

$start\_p$: Start Pixel is an integer number signifying the pixel no. from which calculations should start from.

end_p: End Pixel signifies the pixel no. up to which calculations must take place.

Selective omission is implemented by limiting these values.

- This function takes two arguments, the g_angle (global Grand Angle) and the current yaw of the camera. The g_angle is the previous yaw value of the camera and is initialized to be 0.
- If g_angle is greater than yaw, it means that the camera was rotated in the CCW direction. Hence, the pixel values are calculated from the LHS. This makes $start\_p$ = 0, because it is the first pixel from the LHS. To calculate end_p, the difference in angle values is calculated. This difference is divided with the angle per pixel value which is $\left(\frac{h_{fov}}{frame\ width}\right)$ = 86/640 = 0.134. Therefore, the end_p can be calculated as (g_angle – yaw)/0.134 and rounding it to the nearest integer number.
- If g_angle is lesser than yaw, it means that the camera was rotated in the CW direction. Hence, the pixel values must be calculated from the RHS. This makes end_p = 640, because it is the last pixel from the RHS. To calculate start, the difference in angle values is calculated. This difference is divided with the angle per pixel value which is $\left(\frac{h_{fov}}{frame\ width}\right)$ = 86/640 = 0.134. Therefore, an intermediate point $inter\_p$ can be calculated as (g_angle – yaw)/0.134 and rounded to the nearest integer number. This value indicates the pixel difference. Subtracting this value from 640, gives the value of $start\_p$
- Cases are added where the difference between g_angle and yaw is 0 in which case the $start\_p$ is made 0 and end_p is made 640.
- Finally, the global g_angle is changed to be the current yaw . This value is used in subsequent iterations

III. Results and Discussion

The results generated by implementing the said method using the sensors mentioned earlier are discussed in this section and are compared with various versions of the method.

A. Comparitive Study

The method was developed using an incremental approach. The various versions that led to the development of the method that is proposed are discussed in this section.

- Pilot - Basic method to generate a map was developed with no capacity to handle point redundancies.
- Thresholding enabled - Thresholding was introduced to the method to prevent redundant entries to the generated map. The map is updated only if the vehicle changes its pose more than the threshold value.
- Selective omission - Introduction of selective omission enabled the method to omit overlapping points on the generated map. This results in a smoother output.

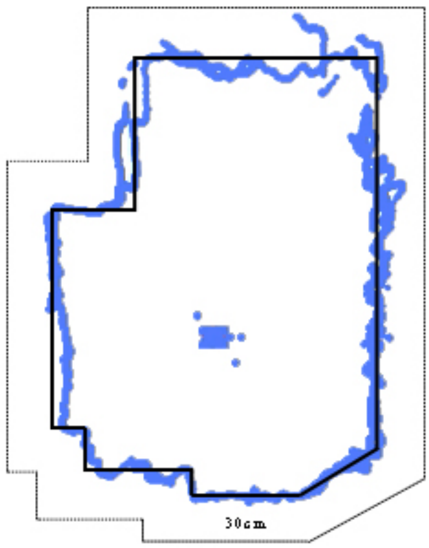

Figure 5 – Approximate blueprint and output stacked

B. *Results*

Fig. 5 demonstrates a map generated by the method proposed in this paper in blue. The black overlay represents a very approximate blueprint of the controlled environment, a room. The results show that the method performs well in areas with optimum lighting conditions. The dotted line outside the stacked boundary shows the extent of the overshoot, i.e. the points generated by the method that are further away from the origin than expected.

Although it seems like the generated output (Fig. 5 Blue) is not as perfect as the blueprint (Fig.5 Black) itself, one has to take into consideration the objects in the room. All the jagged edges and discontinuities observed are generated due to obstacles and objects present in the room. As the aim of the method is to generate a knowledge base, that is, a map that can be used to navigate the environment. It is not practical to ignore these discontinuities as that might lead to omission of certain obstacles in the environment.

C. *Constraints and Limitaions*

- Lighting Conditions - The method performs well under moderate lighting conditions, extreme differences in exposure can result in discontinuities in the output generated by the method
- Vibrations – The method is not capable of handling vibrations that affect the data while it is captured by the sensor. A stable hardware platform is required in order to achieve optimal results.
- Distortions – The method relies on the sensor used to capture data for distortion correction. Pincushion distortions on the extreme ends of the depth frame were observed. Hence, distortion correction must be applied to the frame before being used.

D. *Future work*

- Exposure Correction – While imaging, if there is an area in the image which is overexposed or underexposed it will cause an error in the ranging data. Thus, a processing technique is required that eliminates this error by either normalizing the exposed area or ignoring the exposed region entirely.
- Selective Omission for Translation- The selective omission has been implemented for rotation about a fixed point. This concept can be extended to include cases where the robot has moved either horizontally or vertically and contains overlapping regions with the previous iteration.
- Selective Removal of Pixels – Manually or automatically pass range of values between which no mapping is generated. The benefit of this feature is the fact that, if there is a requirement to ignore certain ranges of pixels, then they must be eliminated despite meeting any prior mapping criteria.
- Extended Awareness - Within the generated pointset, we must be able to determine not only the existence of a potential obstacle and localize it within the environment but also classify and assign a weight to it based on the probability that the object is likely to move making its localization in the environment volatile. E.g. Small furniture like chairs have a non-trivial probability of changing their position in a room whereas storage cabinets are unlikely to move in the short run. This can be used to be aware of humans in the environment and safely operate alongside them without posing as a threat.

IV. Conclusion

This paper proposes a robust, lightweight method for use onboard UAVs and UGVs, that is capable of generating a map of an indoor environment using simple, off-the-shelf sensors and geometric primitives. The paper discusses the efficacy of the method proposed and provides results generated by implementing the method in a controlled environment.

The results demonstrated show that the method can be used in real-life scenarios and is able to generate a fairly accurate map of the environment in the test case while still having a scope for further development. As discussed in section (III-D), the implementation of extended awareness is an important feature as that makes the robot capable of anticipating changes and taking them into account while navigating. Also, different exposure correction methods need to be tested to make the method viable in non-uniform lighting conditions.

Acknowledgment

We would like to thank our institute Fr. Conceicao Rodrigues College of Engineering for the funds and access to the laboratory equipment. We also thank Dr. Varsha Turkar of Don Bosco College of Engineering Goa for her guidance.

References

[1] J. Vandorpe, H. Van Brussel and H. Xu, "Exact dynamic map building for a mobile robot using geometrical primitives produced by a 2D range finder," Proceedings of IEEE International Conference on Robotics and Automation, Minneapolis, MN, USA, 1996, pp. 901-908 vol.1.

[2] Morris, William & Dryanovski, Ivan & Xiao, Jizhong. 3D Indoor Mapping for Micro-UAVs Using Hybrid Range Finders and Multi-Volume Occupancy Grids (2010).

[3] Hiroaki Seki, Yasuo Tanaka, Masaharu Takano, Ken Sasaki, "Positioning System for Indoor Mobile Robot Using Active Ultrasonic Beacons", IFAC Proceedings Volumes, Volume 31, Issue 3, 1998, Pages 195-200.

[4] Y. Park, "Smartphone based hybrid localization method to improve an accuracy on indoor navigation," 2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Busan, 2014, pp. 705-708.

[5] A. Satan, "Bluetooth-based indoor navigation mobile system," 2018 19th International Carpathian Control Conference (ICCC), Szilvasvarad, 2018, pp.