COMPARISON OF UAV LIDAR ODOMETRY OF ROTATING AND FIXED VELODYNE PLATFORMS

Mohamed Lamine Tazir[1], Nicolas Seube[1]

[1]Geown France, 13 avenue de l'Europe, 31520 Ramonville St-Agne (mohamed.tazir, nicolas.seube)@geown.io

KEY WORDS: LiDAR odometry, mapping, rotating Velodyne, UAV, spatial analysis, point clouds comparison.

ABSTRACT:

Three-dimensional LiDAR rangefinders are increasingly integrated into unmanned aerial vehicles (UAV), due to their direct access to 3D information, their high accuracy and high refresh rate, and their tendency to be lightweight and cheaper. However, all commercial LiDARs can only offer a limited vertical resolution. To cope with this problem, a solution can be to rotate the LiDAR on an axis passing through its center, adding an additional degree of freedom and allowing more overlap, which significantly enlarges the sensor scope and allows having a complete spherical field of view (FOV). In this paper, we explore this solution in detail for drone's context, while making comparisons between the rotating and fixed configurations for a Multi-Layers LiDAR (MLL) of type Velodyne Puck Lite. We investigate its impact on the LiDAR Odometry (LO) process by comparing the resulting trajectories with the data of the two configurations, as well as, qualitative comparisons, of the resulting maps.

1. INTRODUCTION

Unmanned Aerial Vehicles (UAV) surveys require accurate positioning. For years, Global Navigation Satellite System (GNSS) has been a widespread complementary solution. However, this system remains sensitive to signal masking and multiple reflections that may occur in dense areas, or in general, anywhere the sky is blocked by obstructions like skyscrapers, dense tree regions, tunnels, or in underground locations.

A prevailing alternative is to use simultaneous localization and mapping (SLAM) algorithms (Cadena et al., 2016), as it is based solely on the information delivered by the drone's own sensors. In this research, and for reasons of compatibility with real-time applications, we will interest in the SLAM front-end part, and only focus on laser technology as it provides direct 3D data, and more precisely, in Multi-layer LiDARs (MLL). These latter allow direct scanning of the environment with a 360°horizontal field of view (FOV), and a vertical opening of several degrees different depending on the model and the number of layers. These LiDARs have experienced a boom in recent decades thanks to their high accuracy and high refresh rate. However, a problem persists with this type of sensor which is their sparsity (Tazir et al., 2018). In other words, the more distant the scanned object is, the greater the spacing between the points reflected by this object. This problem leads to the loss of environment details, and more seriously, can cause LiDAR odometry (LO) crash if there is not enough overlap between scans (Velas et al., 2016). Moreover, depending on the layout of the MLL in the drone (vertical or horizontal), parts of the environment can be omitted. Decisively, their main disadvantage lies in their limited vertical FOV (Zhen, Scherer, 2018). A solution can be to rotate the MLL on an axis passing through its center, adding an additional degree of freedom and allowing more overlap. Indeed, the rotation significantly enlarges the sensor scope and allows having a complete spherical FOV but a more complicated scanning pattern. In this paper, we explore this solution in detail while making comparisons between the rotating and fixed configurations for an MLL (Velodyne Puck Lite). We investigate its impact on the LO process by comparing the resulting trajectories with the data of the two configurations, as well as, qualitative comparisons, of the resulting maps.

2. RELATED WORKS

SLAM is a very broad field and results of more than 30 years of research activity. However, in the drone industry, the interest in this solution, to improve/replace mapping with GPS georeferencing, is relatively recent; it is very novel when it comes to the SLAM applied to the drone's flight in GNSS denied environment using a rotating LiDAR. Indeed, an Australian research team has begun this research area by producing a 1.8 kg payload specially designed for drone platforms (Jones et al., 2019). This payload consists of a rotating MLL, an inertial measurement unit (IMU), and an onboard computer. It uses mainly Velodyne VLP-16 rotating at a 0.5 Hz rate, in conjunction with a proprietary SLAM solution to generate 3D point clouds without the need for GNSS. In mobile robotics, in 2016, Neumann and his colleagues present in (Neumann et al., 2016) a platform for a mobile robot that continuously rotates a Velodyne VLP-16 PUCK together with a Hokuyo UTM-30LX-EW. They claim that they obtain more uniformly distributed point clouds than with a fixed sensor while achieving almost complete sphere coverage of the environment, only a cone of about 71° towards the base of the sensor cannot be acquired with the Velodyne, the Hokuyo's occlusion is even lower. In (Zhen, Scherer, 2018) a rotating VLP-16 with a motor speed of 30 rpm is tested. Their goal was focused on improving SLAM algorithms to manage different types of LiDAR payload, including rotating 3D LiDAR. They conclude that the use of rotating LiDAR provides wider FOV, which could significantly improve the mapping performance. (Pfrunder et al., 2017) uses a Dynamixel MX-106R servomotor to oscillate a Velodyne Puck between -30 and +30 degrees with a frequency of 70 degrees per second. This system is used for the real-time localization of an autonomous ground vehicle. With this configuration, the FOV is expanded, which leads to more robustness and precision in the localization process. The results show that more than 60 km were carried out in completely autonomous driving. All this research addresses the case of the rotary MLL payload partially with different objectives. Our study differs from what was pre-

viously mentioned in the fact that it interests to the impact of the addition of the rotation mechanism on the lidar odometry process for UAV applications. It highlights the impact of the different parameters on the drone's trajectory and the final obtained map.

3. GENERAL FORMULATION

3.1 Rotation mechanism

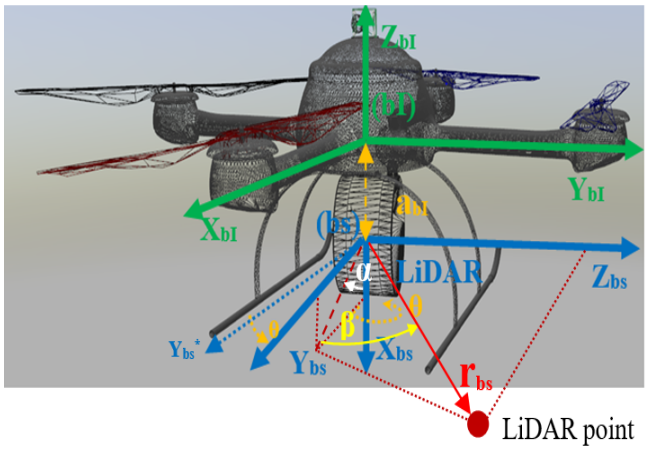


Figure 1. The sensor configuration with reference frames

In our application, the Velodyne rotates around an axis that passes through its center. let us define the LiDAR frame (bs) (X_{bs},Y_{bs},Z_{bs}), centered at its optical center as it is shown in Figure 1. For a given LiDAR point, it is referenced in Cartesian coordinates by (x_{bs}, y_{bs},z_{bs}) and in spherical coordinates by (r_{bs}, α, β) with r_{bs} is the radius and α and β are the two angles following the axes X_{bs} and r_{bs} respectively. Let us also define the frame (bI) (X_{bI}, Y_{bI}, Z_{bI}) attached to the drone and called body frame. If an IMU is used, this frame coincides with the IMU frame (in this work, we do not use an IMU). For this frame, its X_{bI} axis is directed towards the front of the drone and its Z_{bI} axis is pointed upwards.

In the case where the Velodyne is in fixed mode (does not rotate), the X_{bs} axis coincides with the Z_{bI} axis, but is directed downwards, the Y_{bs} is parallel to the X_{bI}, and the Z_{bs} is parallel to the Y_{bI}. This configuration is denoted by ($X_{bs}{}^*$, $Y_{bs}{}^*$,$Z_{bs}{}^*$). In this case, the Cartesian coordinates of a LiDAR point in the drone's frame will be:

$$x_{bI} = r_{bs}\sin(\beta) \tag{1}$$

$$y_{bI} = r_{bs}\cos(\beta)\sin(\alpha) \tag{2}$$

$$z_{bI} = -r_{bs}\cos(\beta)\cos(\alpha) - a_{bI} \tag{3}$$

In the case where the Velodyne rotates around an axis passing through its center (X_{bs}, Y_{bs}), the situation gets more complicated. It should be noted that the rotation around the Z_{bs} axis is unnecessary, as it coincides with the rotation of the LiDAR's beams. In our use case, the Velodyne rotates around the X_{bs} axis with a constant rotational speed. Let us concede θ the rotation angle. The Cartesian coordinates of the LiDAR point in this case will be:

$$x_{bI} = r_{bs}[\cos(\beta)\sin(\alpha)\cos(\theta) - \sin(\beta)\sin(\theta] \tag{4}$$

$$y_{bI} = r_{bs}[\sin(\beta)\cos(\theta) + \cos(\beta)\sin(\alpha)\sin(\theta)] \tag{5}$$

$$z_{bI} = -r_{bs}\cos(\beta)\cos(\alpha) - a_{bI} \tag{6}$$

3.2 LiDAR odometry and mapping

The LiDAR odometry and mapping process (Tazir, 2018) aims to estimate the pose of the drone and the map of the environment at the same time by adjusting the point clouds of its onboard LiDAR sensor. From the received scans, a registration process (Tazir, 2018) is performed between every two successive scans. Commonly, registration is based on two main steps: a correspondence between the points of the two scans to obtain a list of matches. Then a pose computing from the obtained matches. This pose represented by its six parameters (three translation and three rotation parameters) represents the local position of the Velodyne sensor. The transformation of this pose into the global coordinate frame gives the absolute position. The map is built from a set of the aligned scans.

3.2.1 Poses optimization We use a mapping and localization technique that is based only on LiDAR data. We follow the framework proposed in the "LOAM" paper (Zhang, Singh, 2014), which is for years has been considered as the state-of-the-art LiDAR estimation method. The drone estimates its position by aligning the received point clouds. For this, a registration process is performed using the Normal Distribution Transform (NDT) algorithm (Magnusson et al., 2007). In order to reduce the drift, a keyframe strategy (Pomerleau et al., n.d.) was implemented. This solution consists in fixing a reference scan and aligning all the incoming scans to this one. We use distance-based criterion to generate the keyframes.

More formally, let consider ${C_i}^{i+1}$ be the transformation between to consequent scans S_i and S_{i+1}acquired at time i and $i+1$ by the Velodyne sensor, and C_c is the transformation between the current scan and the previous keyframe. The displacement between these two scans can be represented by the 6 DoF vector: $[t_x\ t_y\ t_z\ r_r\ r_p\ r_y]^T$

As shown in Figure 2, the new scan S_{i+1} is registered against the previous keyframe using the transformation C_c which is calculated from the previous transformations C_i^{i+1}, C_{i-1}^{i},..., C_{i-4}^{i-3} as:

$$C_c = (C_{i-4}^{i-3})^{-1} * (C_{i-3}^{i-2})^{-1} * (C_{i-2}^{i-1})^{-1} * (C_{i-1}^{i})^{-1} * (C_i^{i+1})^{-1} \tag{7}$$

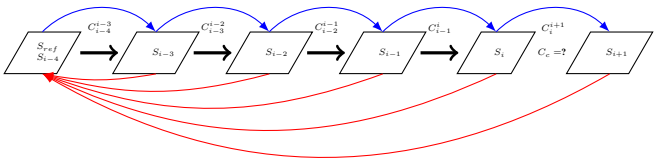


Figure 2. Every incoming scan S_{i+1} is registered against the previous keyframe S_{ref}, and the multiples transformations $C_c^i, C_c^{i+1}, \ldots, C_c^{i+n}$ (red edges) transforming these scans in the coordinate frame of $S_{ref}^i, S_{ref}^{i+1}, \ldots, S_{ref}^{i+n}$ are estimated.

3.2.2 Map construction The map consists only of keyframes. This technique allows both to reduce the error drift and to prevent the creation of keyframes (adding new data to the map) if the drone remains in the same position. This reduces

the information contained in the built-in map by not overloading it with unnecessary information. The map-building process consists of two main steps:

- selection of keyframes from the stream of current clouds,
- update the map from the selected keyframes,

3.2.3 Arrangement with rotational constraint For the case with the LiDAR rotation, a module that retrieves the engine's rotation angle equivalent to each LiDAR scan is implemented. This angle is used to transfer each point cloud from the rotating frame (bs) to the body frame (bI) using the equations (4, 5 , and 6).

Algorithm 1: LiDAR odometry and mapping

Input: scan $S_1, \ldots, S_n$
Output: map, currentPosition
Initialize: $S_{Ref}, S_i, C^n_{bI} = C^{bs}_{bI} = C_{Identity}$
begin
  //Retrieve the initial position and orientation of the drone
    $C^n_{bI} = getInitialposition()$
    $currentPosition = C^n_{bI}$
  // Create the initial transformation matrix
    $C^n_{bs} = C^n_{bI} C^{bs}_{bI}$
  //Transform the first scan from the LiDAR frame to the world frame
    $S_i = transformePointCloud(S_1, C^n_{bs})$
  // Add Initial point cloud to the map
    $map = S_{Ref} = S_i$
  for $i = 2 \quad to \quad N$ **do**
    $S_i = transformPointCloud(S_{i-1}, C^n_{bs})$
    // Registration: Align the current cloud to the reference cloud
      $C^{ref}_i = register(S_i, S_{Ref}, C^{bs}_{bI})$
    // Calculate the sensor pose in the world-based frame
      $C^{bs}_i = C^{bs}_{bI} C^{ref}_i$
      $C^n_{bs} = C^{bs}_i \times C^n_{bI}$
    // Update the position and the orientation of the drone
      $currentPosition = getXYZ\ RPY(C^n_{bs})$
    // Calculate the traveled distance between the current position and the reference position
      $dist = getDistance()$
    if $(dist >= Dist_{thre})$ **then**
      $\bar{S}_i = transformPointCloud(S_i, C^n_{bs})$
      // Insert the Keyframe in the map
        $map+ = \bar{S}_i$;
      // Change the reference cloud by the current cloud
        referenceCloud = currentCloud ;
        $S_{Ref} = S_i$
    else
      // Stay holding the previous reference cloud and match every incoming cloud against it.
    end
  end
  return *map, currentPosition*
end

4. COMPARISON METHODOLOGY

In this section, we will expose the experiments that we want to perform in order to compare the rotating and fixed configurations for an MLL of type Velodyne Puck Lite attached to an md4-1000 drone (Microdrones, 2020). A simulation-based methodology is used to analyze the impact of the rotation on the LO process by comparing the resulting trajectories with the data of the two configurations, as well as, qualitative comparisons, of the resulting maps.

4.1 Tests protocols

For our work, we choose to use Gazebo simulator (Koenig, Howard, 2004) with the Robot Operating System (ROS) (Quigley et al., 2009). These tools allow a high-fidelity simulation, with physical models very close to reality and therefore robust. The drone md4-1000 from the company Microdrones (Microdrones, 2020) was simulated with characteristics very close to the real drone. Then we simulated the Velodyne sensor with the rotation mechanism. Figure 3 shows the simulated drone as well as an urban-like environment that we created for our tests.

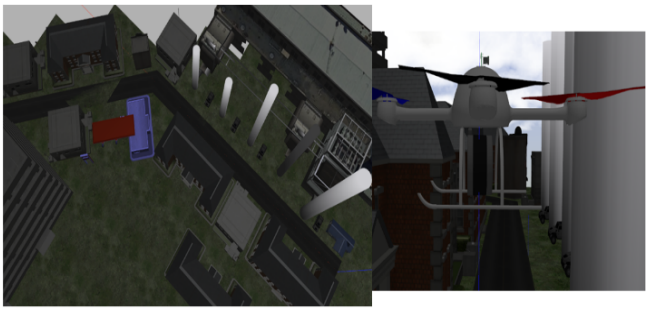


Figure 3. The Microdrones md4-1000 drone simulated in the gazebo simulator with the test environment

The testing protocol consists of flying the drone with the two payloads (fixed Velodyne and rotating Velodyne) and used our LOAM technique to compute drone's trajectories and create the resulting maps. For each test, we vary one of the parameters that influences the LOAM process:

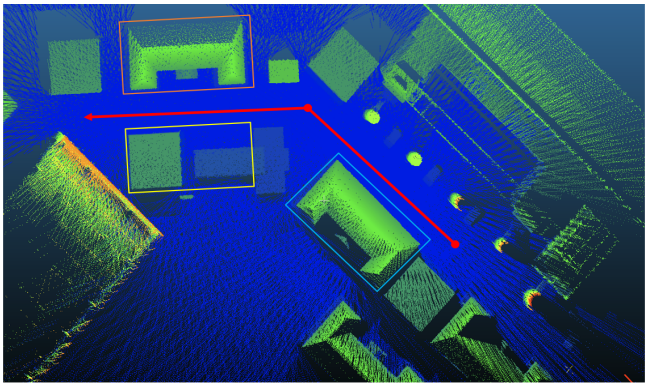
- Experiment 1: trajectories comparison: for this experiment, the drone flies at a height of 30 meters and for a distance of 100 m. The drone's trajectory is shown by the red line on figure 4. Regarding the rotating payload, the Velodyne rotates around the (X_{bs}) axis as shown in figure 1, with a rotation speed of 0.5 rad/s. We quantitatively compare the two resulting trajectories against a ground truth trajectory given by perfect IMU and GPS.
- Experiment 2: resulting maps comparison: we qualitatively compare the rendered maps of the two payloads
- Experiment 3: rotation speed variation: in this test, we vary the Velodyne's rotational speed, and we compare the impact of this variation on the LO results.
- Experiment 4: we test both payloads on other types of environment. This test is performed on real data collected by a real drone.

5. OBTAINED RESULTS

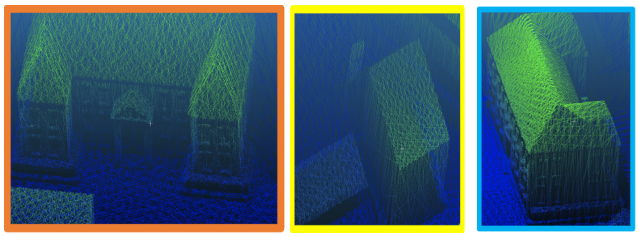
This section presents the obtained results on each of the experiments presented in the previous section.

5.1 Results with rotating Velodyne

The mapping result given by LiDAR Odometry with the rotating payload is shown in the figure 4. In this experiment, the drone flies at a height of 30 meters with a speed of 1 m/s. Its trajectory is shown by the red line on this figure. The total number of points in the resulting map is 2 243 685 points.



(a) The resulting LiDAR Odometry map with the rotating payload



(b) (c) (d)

Figure 4. Mapping results given by LiDAR Odometry with the rotating payload. (b), (c) and (d) are exploded views of the rendered map

These results show:

- patterns due to the combination of the beams are visible.
- walls are well recovered as well as the horizontal surfaces. with a dense overlap for the horizontal walls.
- density (isotropy) increases notably around the intersection with the rotation axis. This appears on the ground points below the drone.

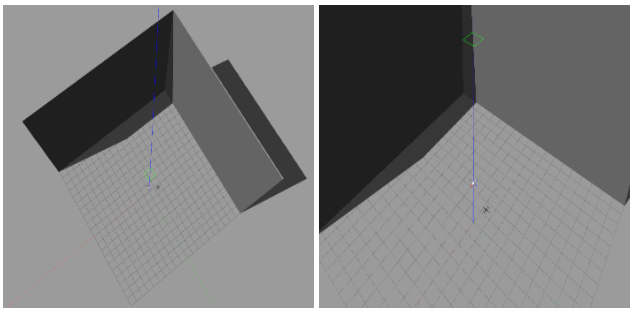
In order to investigate these results precisely, we placed the drone in front of two walls at a distance of 10 m each. The drone is static (does not move) and the Velodyne is in horizontal mode and rotates around the axis (X_{bs}). One of the walls is perpendicular to the Velodyne's rotation axis and the other is parallel to this axis. This is shown in figure 5. In this figure, the (X_{bs}) axis is shown in red, (Y_{bs}) in green and (Z_{bs}) in blue.

This experiment is used to measure the impact of the rotation on perpendicular and parallel walls. The results of this test are shown in figure 6.

The results of this experience are:

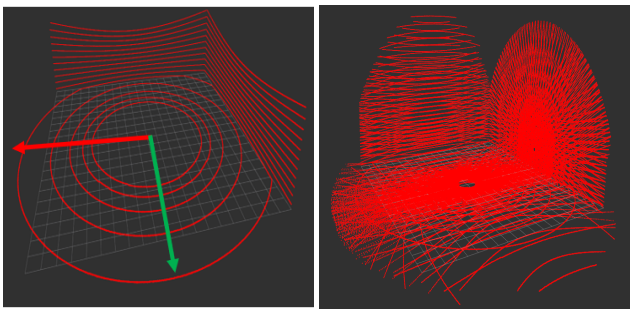
For the Fixed payload, we notice:

- sparse information about the ground



(a) (b)

Figure 5. Experiment to measure the impact of the rotation on perpendicular and parallel walls



(a) Fixed payload (b) rotating payload with 0.5 rad/s

Figure 6. Comparison between fixed and rotation payload on perpendicular and parallel walls

- restricted vertical resolution

For the rotating payload:

- denser data for both ground and walls
- wider FOV
- point patterns on the perpendicular planes depend on the sensor orientation: i.e. the wall perpendicular to the rotation axis (the wall on the right of the figure) is scanned with a higher density in the center. While the wall parallel to the rotation axis (the wall in front) is sampled with a slightly higher density on the sides.
- density increases notably around the intersection with the rotation axis

5.2 Results with fixed Velodyne

The LO module fails to build the map properly. This appears clearly on the trajectories comparison where we see the error drift diverges very quickly after a few meters.

5.3 Trajectories comparisons

This experiment consists of comparing the two trajectories given by the LO with the ground truth trajectory computed from the data of perfect simulated GPS and IMU. This consists of comparing the three translations (positions X, Y, and Z over time) and the three rotations (roll, pitch, and yaw). In addition, two metrics are used for the evaluation, the relative translational error (RTE), and the relative rotational error (RRE). The former

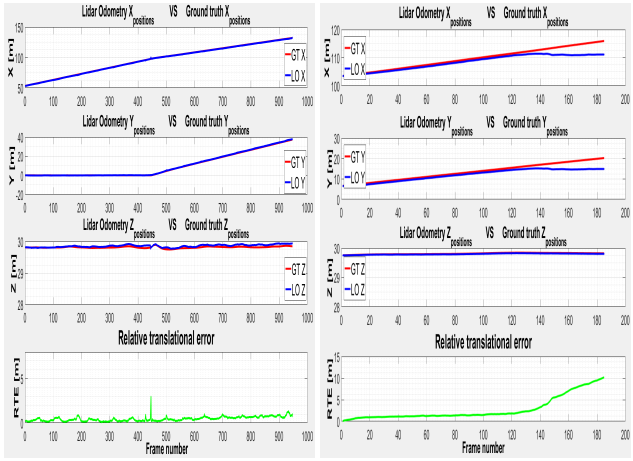
describes the translation gap between the ground truth (t_{GT}) and the estimated (t_E) translation vectors.

$$RTE = \|\mathbf{t}_{GT} - \mathbf{t}_E\|_2 \tag{8}$$

The second metric is the Relative Rotational Error (RRE), we use the metric defined on the tangent space of $\mathbb{SO}(3)$:

$$RRE = \| \log\mathrm{m}(\mathbf{R}_E^T \mathbf{R}_{GT})\|_F \tag{9}$$

where $\log\mathrm{m}(.)$ is the matrix logarithm, $\mathbf{R}_E$ is the estimated rotation matrix, $\mathbf{R}_{GT}$ is the ground truth rotation matrix and $\|.\|_F$ is the Frobenious norm. In the ideal case when the rotation matrix $\mathbf{R}_E$ is exactly equal to $\mathbf{R}_{GT}$, the multiplication $\mathbf{R}_E^T \mathbf{R}_{GT}$ would be the identity matrix, and thus the rotational error equal to zero.



(a) rotating payload with 0.5 rad/s (b) Fixed payload

Figure 7. Comparison between fixed and rotation payload on LiDAR odometry positions

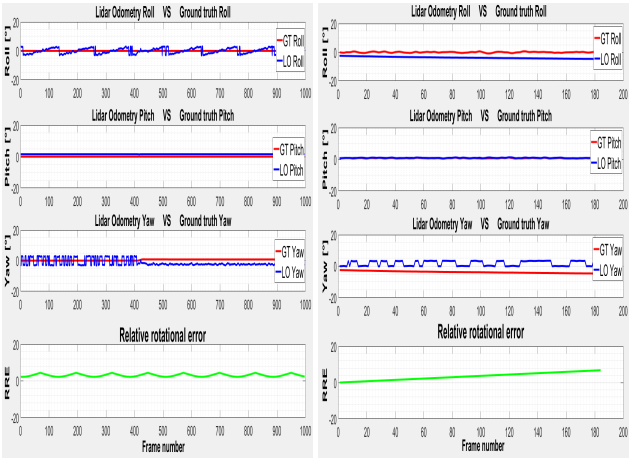
Figures 7 and 8 show the comparison between the pose estimation of the LiDAR odometry with the rotating payload and with the fixed payload (blue line). The ground truth is given in red line. These figures present the localization accuracy in terms of x, y, and z positions relative to the ground truth, as well as the relative translational error (RTE) and the roll, pitch, and yaw orientations with the relative rotational error (RRE).

Results of the LIDAR Odometry on the rotation payload data:

- Despite a small spike at the time of trajectory change (drone imbalance), around 45 m, we have a good alignment for the three positions x, y, and z, throughout the trajectory.
- A drift of 1m over all the three positions (more visible on the RTE graph) begins to appear around of 100 m of trajectory.

For the Fixed payload, we notice that:

- The LOAM module fails to build the map properly. It spits after 20 meters, because of a large cumulative drift on the three positions that reaches the 10 m.



(a) rotating payload with 0.5 rad/s (b) Fixed payload

Figure 8. Comparison between fixed and rotation payload on LiDAR odometry orientations

- The drift is greater on the two horizontal components (X and Y). This is due to the restricted FOV (as the Velodyne is in a fixed vertical configuration (directed downwards) and with a flight height of 30 m).

Regarding the orientations, for the rotating payload, the RRE remains little ready steady. Unlike the fixed payload which diverges very quickly.

These results show that the performance achieved by the LO on the rotation payload data is greater than that with the fixed payload.

5.4 Resulting maps comparisons

The purpose of this test is to compare the LOAM resulting map of the rotating payload data shown in figure 4, with the map created by geo-referencing the LiDAR data of the Velodyne Puck in fixed mode using the perfect simulated GPS and IMU data. Similar as we compared the LO trajectory with the ground truth trajectory, we compare the resulting map with a supposedly perfect map. This last map is shown in the figure 9.

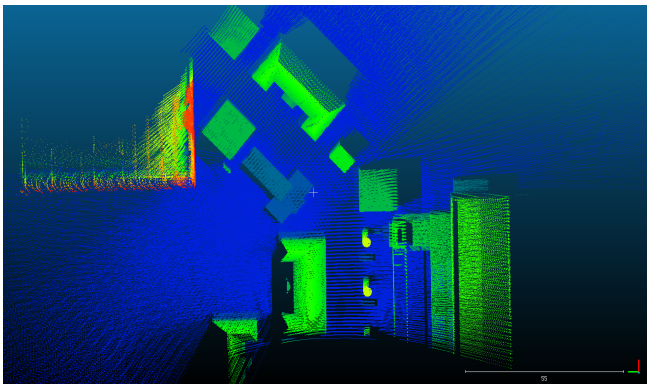


Figure 9. Map produced by georeferencing the LiDAR data of the fixed Velodyne Puck

In order to compare the two maps, we have superimposed them in CloudCompare (CloudCompare, 2020) (Figure 10). In this figure, the georeferenced map is shown in blue, while the map created by our LOAM technique using rotating payload data is

in red. Both maps are the results of the same trajectory executed by the drone equipped each time by one of the two payloads.

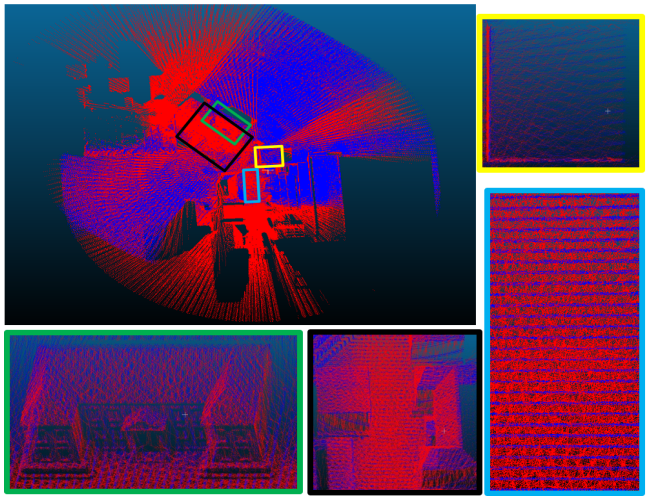


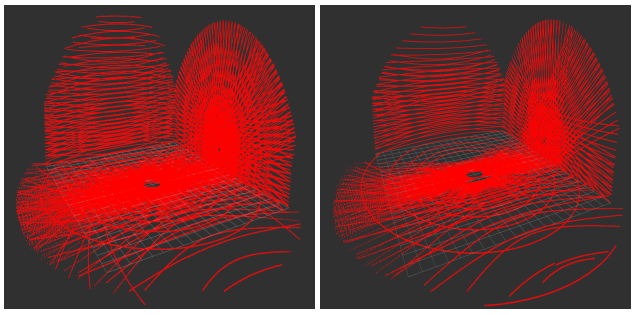
Figure 10. Comparison between the two rendered maps: fixed (bleu)and rotating payload (red). Images surrounded by colors show zooming parts shown by squares of the same color on the main figure

This experience reveals the following observations:

- good overlap between the two maps
- very good recovery (isotropy) in the case of rotating payload
- glaring spaces on the horizontal part of the environment for the case of the fixed payload
- lack of overlap on vertical walls for the fixed payload
- more scope for the rotating payload

5.5 Varying the rotation speed

We started by testing the speed variation in a static mode where the drone does not move, with the environment of the two walls. Then we performed the test on the urban environment with the moving drone. The results of the first test are given by the figure 11.



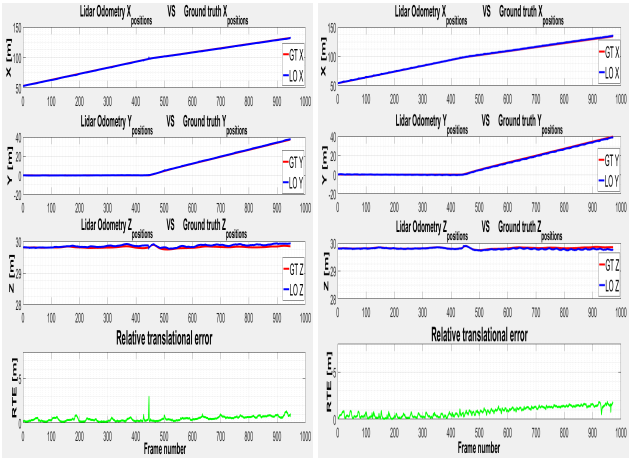
(a) rotating payload with 0.5 rad/s (b) rotating payload with 1 rad/s

Figure 11. Varying the payload rotation speed in a static drone mode

by varying the rotation speed, we note:

- The slower scan offers more density (isotropy): i.e. the band on the ground is wider for the rotation speed of 0.5 rad/s (30 degrees) indicates a higher density for this slower speed compared to the scan of 1 rad/s (60 degrees/s).

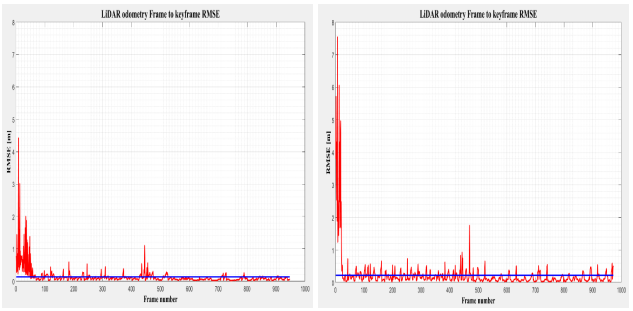
For the second experiment (moving drone in urban environment), we keep all the parameters unchanged, we only change the payload rotation speed. Figure 12 shows the results of this experiment.



(a) Payload rotation speed 0.5 rad/s (b) Payload rotation speed 1 rad/s

Figure 12. Experience of varying the payload rotation speed in a moving drone mode

In accordance with the static drone mode, the results of this experiment show that the slower rotational speed is better for the LO. Because with slower rotation payload, we get more overlap than with a fast rotation speed, which contributes to more density (isotropy) leading to precise registrations processes between successive scans (figure 13). This implies an accurate LO. These results appear well on RTE graphs, with an average of 0.4666 m along the full trajectory for the payload rotation speed of 0.5 rad/s versus 0.8009 m for the rotational speed of 1 rad/s.



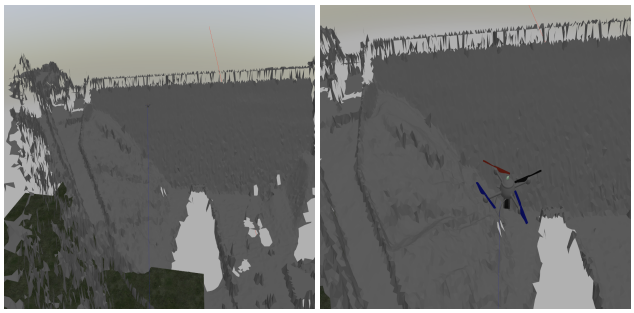
(a) Payload rotation speed 0.5 rad/s (b) Payload rotation speed 1 rad/s

Figure 13. Registration error between successive scans by varying the payload rotation speed

Figure 13 shows the root-mean-square-error evolution of the registration process for both 0.5 rad/s and 1 rad/s speeds. The average is given by the blue line on this figure. This average is 0.1353692 m for the 0.5 rad/s rotation speed and 0.2231537 m for the 1 rad/s.

5.6 Tests on other types of environments

5.6.1 Guerledan dam environment For this test, we have used the data taken during a real survey on the Guerledan dam, carried out by md4-1000 drone and the mdLiDAR1000 sensor. This data was used to generate its surface model. The latter was adapted and imported into the Gazebo simulator. Figure 14 shows the results of this simulation. In this test, the drone performs the same mission executed by the real drone (same trajectory, flight height of 70 m and speed of 4 m/s), except the used payload. The latter is the Velodyne Puck Lite.



(a) (b)

Figure 14. simulation with real data from Guerledan dam

The result of the LOAM process on this environment with the rotating payload is given by figure 15. Whereas for the fixed payload and with the same characteristics (speed of 4 m/s, flight height 70 m) the LOAM is unable to create a correct map of the environment.

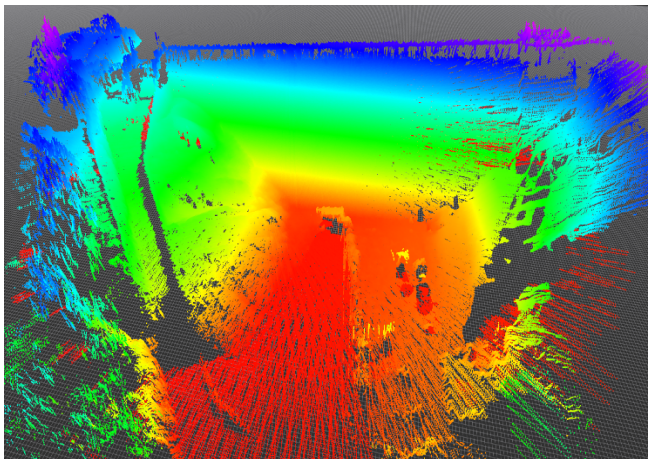


Figure 15. Map of the Guerledan dam generated by LOAM process on rotation payload data

6. CONCLUSION

In this research, we investigate the impact of the payload rotation on LiDAR Odometry and Mapping process by comparing it with the results of the fixed payload. We chose a Velodyne Puck Lite LiDAR, for its lightweight, its affordable price, and its symmetrical FOV. We compared the results of the two payloads, qualitatively and quantitatively, on the drone's trajectory and the final obtained map.

Different conclusions are drawn:

- the rotation significantly enlarges the sensor scope and allows having a complete spherical FOV.
- LiDAR rotation benefits the LiDAR Odometry and Mapping process and offers greater performances than the fixed payload.
- good coverage for the rotating payload, whether for vertical or horizontal surfaces.
- density (isotropy) on the perpendicular surfaces depend on the sensor orientation.
- slower rotation speed offers more density(isotropy), and therefore it is better for LO.

All these points must be taken into account when deciding on the orientation and the rotation axis of the sensor attached to the drone in relation to the target surfaces in a particular application.

REFERENCES

Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I., Leonard, J., 2016. Past, Present, and Future of Simultaneous Localization And Mapping: Towards the Robust-Perception Age. *IEEE Transactions on Robotics*, 32(6), 1309–1332.

CloudCompare, 2020. 3D point cloud and mesh processing software. CloudCompare. http://www.cloudcompare.org/ (1 May 2020).

Jones, E., Sofonia, J., Canales, C., Hrabar, S., Kendoul, F., 2019. Advances and applications for automated drones in underground mining operations. *The Southern African Institute of Mining and Metallurgy*, 323(June), 24–25.

Koenig, N., Howard, A., 2004. Design and use paradigms for gazebo, an open-source multi-robot simulator. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sendai, Japan, 2149–2154.

Magnusson, M., Lilienthal, A., Duckett, T., 2007. Scan registration for autonomous mining vehicles using 3D-NDT. *Journal of Field Robotics*, 24(10), 803–827.

Microdrones, 2020. The md4-1000 drone. Microdrones. https://www.microdrones.com/en/integrated-systems/mdlidar/mdlidar1000-aas/ (1 May 2020).

Neumann, T., Dülberg, E., Schiffer, S., Ferrein, A., 2016. A rotating platform for swift acquisition of dense 3D point clouds. *Proceedings of the International Conference on Intelligent Robotics and Applications*, 257–268.

Pfrunder, A., Borges, P. V., Romero, A. R., Catt, G., Elfes, A., 2017. Real-time autonomous ground vehicle navigation in heterogeneous environments using a 3D LiDAR. *IEEE International Conference on Intelligent Robots and Systems*, 2601–2608.

Pomerleau, F., Magnenat, S., Colas, F., Liu, M., Siegwart, R., 2011. Tracking a Depth Camera : Parameter Exploration for Fast ICP. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1–7.

Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Berger, E., Wheeler, R., Mg, A., 2009. ROS: an open-source Robot Operating System. *ICRA Workshop on Open Source Software*, 1–6.

Tazir, M. L., 2018. Precise localization in 3D prior map for autonomous driving. PhD thesis, UNIVERSITÉ CLERMONT AUVERGNE.

Tazir, M. L., Gokhool, T., Checchin, P., Malaterre, L., Trassoudaine, L., 2018. CICP: Cluster Iterative Closest Point for sparse–dense point cloud registration. *Robotics and Autonomous Systems*, 108, 66–86. https://doi.org/10.1016/j.robot.2018.07.003.

Velas, M., Spanel, M., Herout, A., 2016. Collar Line Segments for fast odometry estimation from Velodyne point clouds. *IEEE International Conference on Robotics and Automation (ICRA)*, 4486–4495.

Zhang, J., Singh, S., 2014. LOAM : Lidar Odometry and Mapping in Real-time. *Robotics: Science and Systems*.

Zhen, W., Scherer, S., 2018. A Unified 3D Mapping Framework using a 3D or 2D LiDAR. *International Symposium on Experimental Robotics*, 1–11.