ENHANCEMENT OF REAL-TIME SCAN MATCHING FOR UAV INDOOR NAVIGATION USING VEHICLE MODEL

S. Zahran [1], A. Moussa [1,2], A. Sesay[3] , N. El-Sheimy [1]

[1] Dept. of Geomatics Engineering, University of Calgary, Calgary, AB, Canada – (Shady.zahran1, elsheimy) @ucalgary.ca
[2] Department of Electrical Engineering, Port-Said University, Port Said, Egypt – amelsaye@ucalgary.ca
[3] Department of Electrical Engineering, University of Calgary, Calgary, AB, Canada – sesay@ucalgary.ca

Commission I, WG I/6

KEY WORDS: Iterative Closest Point (ICP), Hector SLAM, Vehicle Model (VM), Scan Matching, Unmanned Aerial Vehicle (UAV).

ABSTRACT:

Autonomous Unmanned Aerial Vehicles (UAVs) have drawn great attention from different organizations, because of the various applications that save time, cost, effort, and human lives. The navigation of autonomous UAV mainly depends on the fusion between Global Navigation Satellite System (GNSS) and Inertial Measurement System (IMU). Navigation in indoor environments is a challenging task, because of the GNSS signal unavailability, especially when the utilized IMU is low-cost. Light Detection and Ranging Radar (LIDAR) is one of the mainly utilized sensors in the indoor environment for localization through scan matching of successive scans. The process of calculating the rotation and translation from successive scans can employ different approaches, such as Iterative Closest Point (ICP) with its variants, and Hector SLAM. ICP and Hector SLAM iterative fashion can greatly increase the matching time, and the convergence is not guaranteed in case of harsh maneuvers, moving objects, and short-range LIDAR as it may get stuck in local minima. This paper proposes enhanced real-time ICP and Hector SLAM algorithms based on vehicle model (VM) during sharp maneuvers. The vehicle model serves as initialization step (coarse alignment) then the ICP/Hector serve as fine alignment step. Test cases of quadcopter flight with harsh maneuvers were carried out with LIDAR to evaluate the proposed approach to enhance the ICP/Hector convergence time and accuracy. The proposed algorithm is convenient for UAVs where there are limitations regarding the size, weight, and power limitations, as it is a stand-alone algorithm that does not require any additional sensors.

1. INTRODUCTION

Recently, small and micro UAV has attracted great attention from researchers, casual users, and corporations because of the wide variety of applications that utilize the UAV to save time, cost, effort and their ability to execute dangerous tasks without exposing human lives to danger. To perform these tasks, the UAV must be able to operate autonomously without human interventions. This could be achieved through accurate vehicle navigation, which mainly depends on GNSS and IMU fusion in the UAVs. GNSS signal availability is not always guaranteed due to blockage, multipath, jamming, or spoofing (Wang et al., 2014). This greatly deteriorates the performance of the navigation system, especially when the onboard IMU is low-cost. In the indoor environment, GNSS signal is completely unavailable which increases the challenge of vehicle localization especially when the environment is completely unknown.

One of the common methods that is commonly used to navigate the UAV during GNSS signal outage in indoor environment, is Simultaneous Localization and Mapping (SLAM). SLAM approaches can employ different sensors such as LIDAR (Schnipke et al., n.d.), cameras (Chugo, Yokota, Matsushima, & Takase, 2010) and ultrasonic sensors (Holm & Nilsen, 2010) to navigate autonomously (Grzonka, Grisetti, & Burgard, 2009). SLAM is the process of estimating the vehicle navigation states (position, velocity, and attitude) and the surrounding map simultaneously using the sensors' observations that describe the surrounding environment (Cadena et al., 2016).

SLAM with LIDAR is mainly utilized in indoor environments, because LIDAR is capable of providing reach information (angle and range measurements) that describe the environment around the moving platform (Gao, Liu, Atia, & Noureldin, 2015). Also, LIDAR is more immune to environmental changes, especially when compared to cameras, as it can work during day and night and it did not need features to work.

One of the main utilized techniques for calculating the vehicle navigation states, or in other words, the current states transformation of the vehicle compared to the last states, is carried out through scan matching (Shu, Xu, & Huang, 2013). Scan matching algorithms figures out this transformation by solving the optimization problem in an iterative fashion. Scan matching is the process of comparing the current scan in case of LIDAR, with the previous scan/map, to find the amount of rotation and translation that has occurred from the last scan to the current scan/map at the current time. The process of comparing the scans (scan matching) can employ different approaches, like ICP (Zhang, 1994) which is one of the most commonly used algorithms with its variants such as point-to-line ICP (PLICP) (Censi, 2008), polar scan matching (PSM) (Diosi & Kleeman, 2007). Other point-to–map and feature-to-feature matching algorithms are also employed such as Hector SLAM (Kohlbrecher et al., 2013), or corner based scan matching (H. Mohamed, Moussa, Elhabiby, El-Sheimy, & Sesay, 2017). The main advantage of the ICP and Hector algorithms that they can work in an unstructured environment without any need for linear or corner features(Ulaş & Temeltaş, 2011). Point to point ICP algorithm can be used in a featureless environment, as it matches the points themselves instead of features (Shu et al., 2013). , In ICP algorithm, each point of the current scan is matched to the nearest point of the previous scan according to their Euclidean distance and this matching helps to iteratively estimate the transformation between the scans. Hector SLAM adopts a point to map matching approach that solves the SLAM problem using occupancy grid, based on nonlinear optimization. The occupancy grid map is filled according to a probabilistic approach based on the Bayesian theorem. The iterative fashion of these algorithms (ICP/Hector) increases time consumption, and it is not guaranteed to converge as it may fall in local minima, especially when there is harsh or fast movement or in the presence of moving objects.

To enhance the matching accuracy and to decrease the number of iterations required of the ICP/Hector SLAM algorithm, features scan matching (at least one feature) is conducted first as an initialization step before the ICP algorithm (H. A. Mohamed et

al, 2016). First, the algorithm attempts to find a feature (corner) in the previous scan and match it to the current one to obtain the initial translation and rotation, followed by ICP algorithm, as a fine-tuning for the matching process. This algorithm provides a great enhancement in the overall solution RMSE and the required number of iterations. However, it requires at least one feature, otherwise, it will achieve the same performance of the standard ICP algorithm.

Shu et al., 2013, merged the feature-based scan matching with a point to line ICP (PLICP) to find the match between point(s) and the corresponding line(s) to which this point belongs. In this feature-based algorithm, the features are categorized into two categories, translation features, and rotation features to detect the transformation between these features in a closed form manner, based on the geometry of the detected feature. The first group (rotational features) includes the lines, as rotation between lines is more observable than the rotation between points, while the second group (translation) contains distinct points, chosen according to predefined criteria. In this experiment (Shu et al., 2013), the authors used HOKUYO LIDAR URG-04LX-UG01 with laptop 2.1 GHz processor and 3GB RAM and compared their algorithm to canonical scan matching (CSM), and High speed and yet accurate indoor/outdoor tracking (HAYAI). Their algorithm showed better performance than the other two algorithms. HAYAI average estimation error was almost 12.3 times greater than their algorithm, although HAYAI was faster than their algorithm. This approach also requires features like lines in order to compute the rotation otherwise it will achieve the same performance as the PLICP.

The authors of (H. A. Mohamed et al, 2017) enhanced the Hector SLAM algorithm by introducing an initialization step through detected corners followed by ICP algorithm. This initialization step diminishes the requirement for multi-resolution Hector SLAM, to small grid size cell single resolution Hector SLAM.

Because of the iterative fashion of these algorithms, they consume more time and are not guaranteed to converge during UAV sharp movement (rotation/translation). The proposed algorithm in this paper addresses the challenging convergence situations such as sharp maneuvers without the need for features and without the use of extra aiding sensors in the matching process.

2. SYSTEM OVERVIEW

The paper proposes an enhanced real-time scan matching approach based on initialization step (coarse alignment), to estimate the rotation of the UAV with the aid of the vehicle model without the need for any features, followed by normal ICP/Hector SLAM algorithm (fine alignment) as shown in Figure 1. This initialization step helps to decrease the required number of iterations needed for convergence, and decrease the Root Mean Square Error (RMSE). The proposed VM initialization step can be utilized to enhance and aid any other scan matching iterative algorithm. The proposed approach was verified with two experiments, that includes sharp rotations, with the aid of 3DR Solo quadcopter drone, equipped with PX4 autopilot, low-cost LIDAR (RPLIDAR A1) and UDOO X86 ULTRA onboard PC.

The proposed approach aims to utilize the VM of the quadcopter, to estimate the amount of rotation exhibited by the quadcopter. It starts by detecting the beginning of harsh rotation movement using motors relations according to a pre-defined threshold, then according to the time taken until the end of this harsh manoeuvre, which is also detected with the aid of VM, and with the aid of gyro reading during this movement, the rotation angle is estimated, and used to transform the current scan (coarse alignment), then the ICP/Hector SLAM algorithm is utilized as a fine alignment step as shown in Figure 1.

It is worth noting that because of the autopilot stabilization, gyro measurements exhibit frequent noisy spikes even if there is no rotation. Therefore, the raw gyro measurements are not suitable for detecting the start of the harsh rotation. While filtered gyro measurements can be used to eliminate the effect of the noisy measurements to detect start/end of the manuever, such detection will exhibit unaccepted long delay. Therefore, vehicle model has been employed for precisely detecting start/end of the harsh rotation.

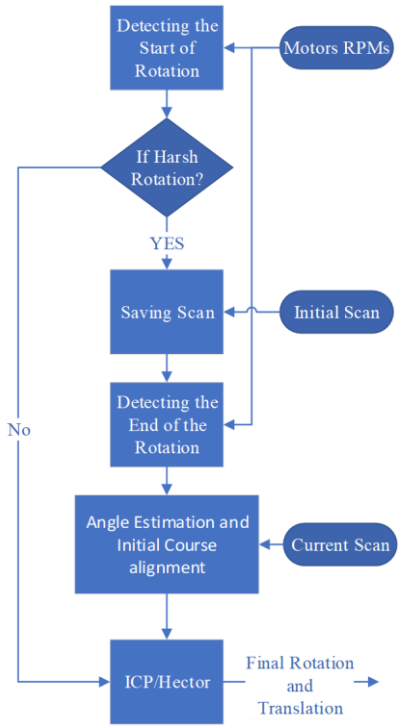

Figure 1. Proposed approach workflow

3. HARDWARE AND QUADCOPTER MODEL

3.1 Vehicle Model

Quadcopter mainly inhabits either plus (+) or X configuration as shown in Figure 2. X configuration is the employed configuration for this research.

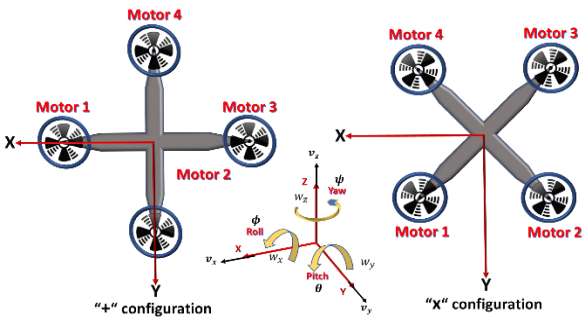

Figure 2. Quadcopter basic configuration

3.2 Basic Quadcopter Motions

Figure 3 shows the 3-basic rotations of the quadcopter (Roll, Pitch, and Yaw). Since the adopted configuration is X, all the motors are responsible for the movements. Roll movement is achieved by increasing the speed of motors 2 and 3 together while decreasing the speed of motors 1 and 4 together and vice versa,

while pitch is achieved through increasing the speed of motors 1 and 2 together and decreasing the speed of motors 3 and 4 together, Finally the Yaw movement is achieved by increasing the speed of motors 1 and 3 together while decreasing the speed of motors 4 and 2 together, and vice versa.

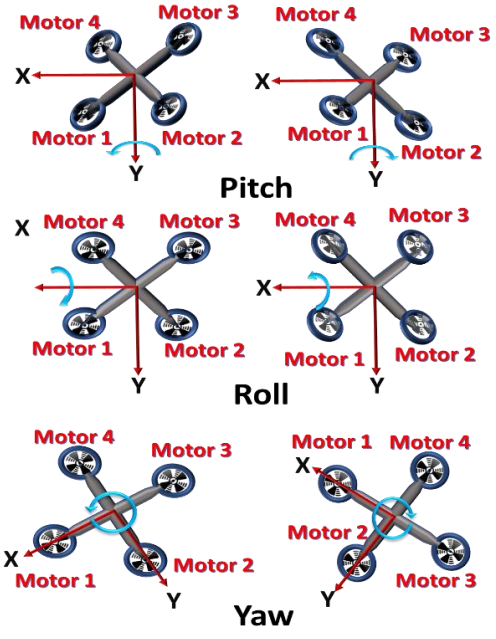

Figure 3. Quadcopter basic rotations

3.3 Hardware Setup

The utilized LIDAR during these experiments is a 2D 360-degree low-cost LIDAR "RPLIDAR A1" of approximately 7-HZ with a maximum detection range of 6 meters. The data is collected with the aid of UDOO X86 which is onboard PC based on Intel quad-core processor with 8GB of RAM. The RPLIDAR and the onboard PC is mounted on quadcopter platform, equipped with 3DR PX4 autopilot based on a MEMS-based IMU (MPU-6000), as shown in Figure 4.

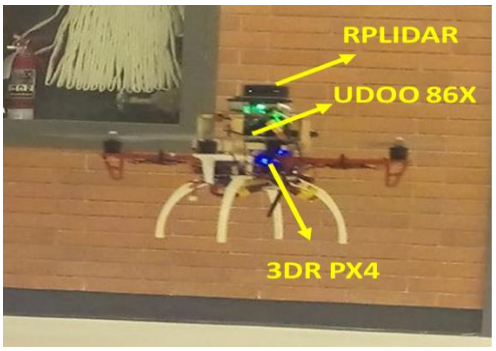

Figure 4. Quadcopter fully assembled with LIDAR, autopilot, and embedded PC

4. PROPOSED METHODOLOGY

The proposed methodology mainly depends on detecting the start and the end of harsh rotations through the vehicle model (motors relations responsible for such movement). According to the speed of motors along the time of this harsh maneuver, an initial estimate of the rotation angle is obtained and used as a coarse alignment to rotate the LIDAR scan at the end of the maneuver to the start scan taken or the constructed map. ICP or Hector SLAM algorithm is used later as fine alignment.

The first step in the proposed approach is how to detect the harsh rotation from the vehicle model. The quadcopter dynamic state that accounts for the rate of change of roll, pitch, and yaw angular rates is shown in Equation 1 (Mahony, Kumar, & Corke, 2012).

$$\dot{W}^b = \begin{bmatrix} \dot{w}_x \\ \dot{w}_y \\ \dot{w}_z \end{bmatrix} = (I^b)^{-1}[M^b_{A,T} - \Omega^b I^b W^b] \quad (1)$$

where w_r is the motor angular velocity, I^b is 3x3 a matrix which accounts for the inertia of the vehicle in the body frame as shown in Equation 2, Ω^b is the skew symmetric form of the body angular rates as given in Equation 3, while W^b is the body rotational rate (angular velocity), finally $M^b_{A,T}$ sums up the all aerodynamic force, torque, and moment resulting from the motors relations as shown in Equation 4.

$$I^b = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & Izz \end{bmatrix} \quad (2)$$

$$\Omega^b = \begin{bmatrix} 0 & -w_z & w_y \\ w_z & 0 & -w_x \\ -w_y & w_x & 0 \end{bmatrix} \quad (3)$$

$$M^b_{A,T} = \begin{bmatrix} dC_T\bar{w}_2^2 - dC_T\bar{w}_4^2 + j_m w_y(\frac{\pi}{30})(\bar{w}_1 - \bar{w}_2 + \bar{w}_3 - \bar{w}_4) \\ -dC_T\bar{w}_1^2 + dC_T\bar{w}_3^2 + j_m w_x(\frac{\pi}{30})(-\bar{w}_1 + \bar{w}_2 - \bar{w}_3 + \bar{w}_4) \\ -C_Q\bar{w}_1^2 + C_Q\bar{w}_2^2 - C_Q\bar{w}_3^2 + C_Q\bar{w}_4^2 \end{bmatrix} \quad (4)$$

According to the previous equations, the yaw angular velocity rate is mainly dependent on the term $M^b_{A,T_{3,1}}$ shown in Equation 5.

$$M^b_{A,T_{3,1}} = C_Q(\bar{w}_1^2 - \bar{w}_2^2 + \bar{w}_3^2 - \bar{w}_4^2) \quad (5)$$

The harsh maneuver is more observable using the derivative of the terms in Equation 5. When the derivative value passes a pre-defined threshold, it's considered the start of the maneuver (this threshold is defined experimentally using the measured speeds of motors during such maneuver in several flights). When the derivative value passes this threshold again it's considered the end of these harsh maneuver as shown in Figure 7. The direction of the rotation is determined based on the sign of the term in Equation 5 (negative peak occurs first in a counter-clockwise rotation, while positive peak occurs first in a clockwise rotation).

5. EXPERIMENTS AND RESULTS

The proposed approach performance and the time consumption is compared with the standard ICP/Hector SLAM algorithm alone without the VM-initialization step in two experiments that include harsh maneuver.

5.1 First Experiment

This first test scenario includes a harsh manoeuvre reached a 2.5 rad/sec rotation in a counter-clockwise direction. This maneuver extended for 1.6 sec with a total number of 11 scan frames taken by the LIDAR.

5.1.1 ICP Performance:

In order to evaluate the performance of the ICP algorithm to be compared with VM-ICP approach, the ICP algorithm is repeated different times, according to the number of frames skipped as shown in Figures 5 and 6, to see the frame skipping effect on the convergence performance and time consumption.

During the experiment, the criteria to stop the ICP algorithm during the matching process is either to perform 100 iterations, or the enhancement in the RMSE from iteration to iteration is less than 10^{-4}.

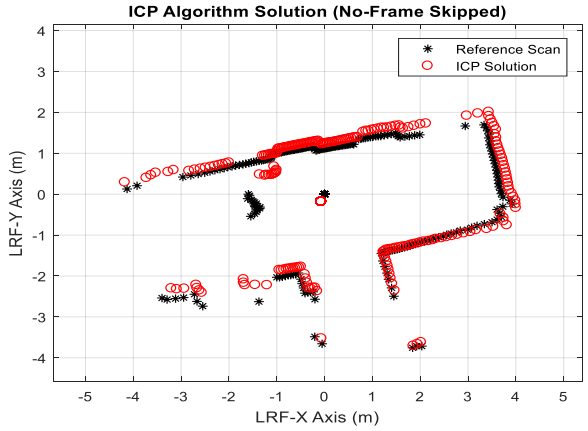

Figure 5. ICP algorithm performance when all frames is used (first Experiment, no frame skipped)

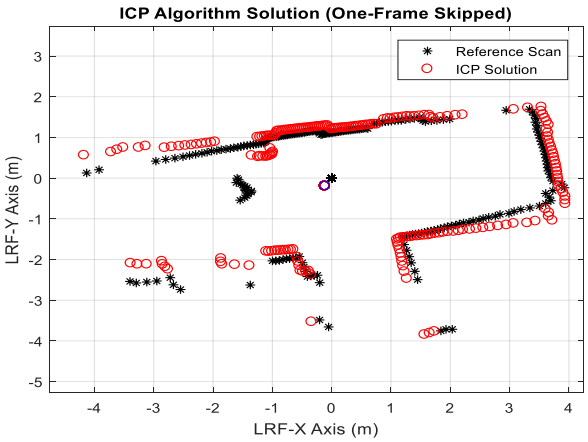

Figure 6. ICP Algorithm performance when one frame is skipped during the harsh maneuver (first experiment, one frame skipped)

Table 1 summarizes the performance of the ICP algorithm, during the first experiment. The experiment shows that during harsh rotation the best RMSE is around 0.21 m with total execution time of 0.39 sec, while the best execution time is 0.062 sec with RMSE of 0.81 m.

Frame Rate	RMSE (m)	TIME (sec)
No Frames Skipped	0.2107	0.0964
Skipping One Frame	0.3108	0.0351
Skipping Four Frames	0.8123	0.0148
Skipping Ten Frames	Stuck in local minima	

Table 1. First Experiment ICP Results

5.1.2 Vehicle Model-ICP:

The Motors-based term shown in Equation 5 is used to detect yaw movement to detect the start and the end of the rotation as shown in Figure 7.

In this experiment, according to the VM-based detection of maneuver duration, the estimated duration of the manoeuvre was 1.6 sec while the actual manuever duration is 1.62 (manually obtained from scans investigation) which indicates an accurate estimation of the manuever duration. taking the reading of the gyro during this period the estimated rotation was 106 degrees. The first detected manuever frame has been rotated based on the estimated rotation angle as course alignment then the ICP algorithm has been applied to obtain the solution shown in Figure 8.

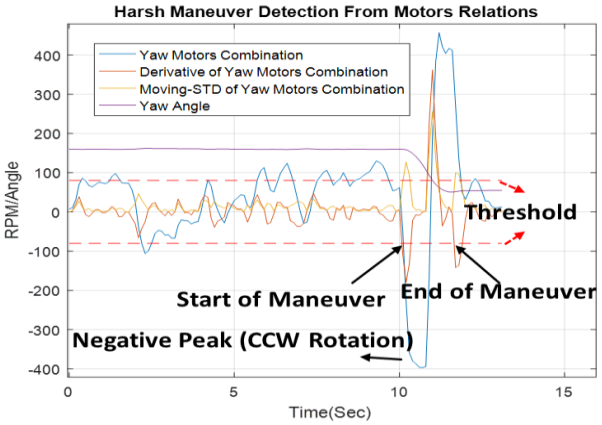

Figure 7. Harsh maneuver detection according to VM motors

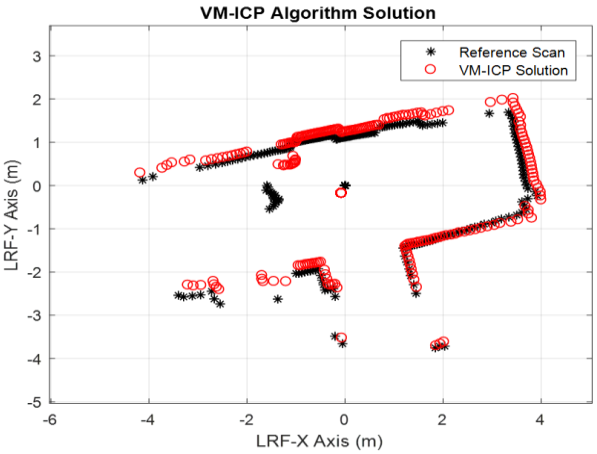

Figure 8. Vehicle-Model ICP performance.

The performance of the proposed approach shown in Table 2 indicates better performance than the typical ICP alone in both RMSE and execution time.

	RMSE	TIME
VM-ICP	0.17 m	0.0023 sec

Table 2. VM-ICP Results (First Experiment)

5.1.3 Hector SLAM Performance (First Experiment):

The Hector SLAM algorithm (H. A. Mohamed et al., 2017) has been implemented for evaluation of its stand-alone version and the assisted version using the proposed approach. The implemented Hector adopts multilevel procedure that begins by solving for grid size 20 cm, then 10 cm followed by 5 cm grid cell. This multilevel procedure aims to overcome falling in local minima that may occur due to gradient ascent. This experiment follows the same procedures presented in the previous ICP experiment. The Hector SLAM has been tested without skipping any frame, then with different step sizes, to see the effect on time consumption and RMSE during harsh maneuver. Figures 9 and 10 show a sample of the experiments, while the full results is shown in Table 3.

Figures 9, 10, and 11 show that the Hector SLAM algorithm was able to converge only when there is no frame skipped, with

RMSE 0.24 m although it required 0.41 sec mean time to register each scan to the constructed map, with total time of 4.5 sec during the whole harsh maneuver.

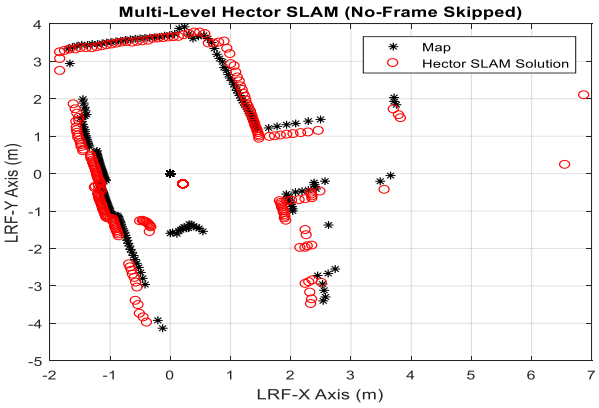

Figure 9. Hector SLAM performances no-frame skipped.

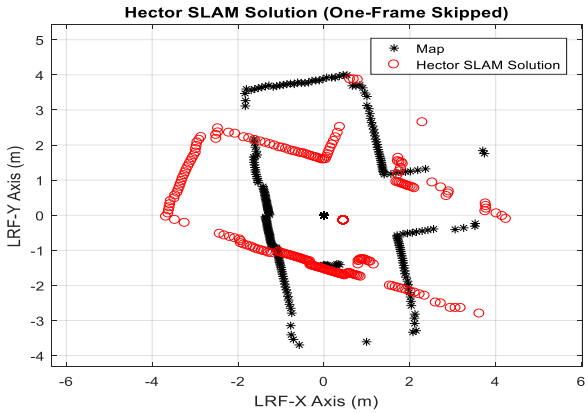

Figure 10. Hector SLAM performances when one-Frame skipping is utilized.

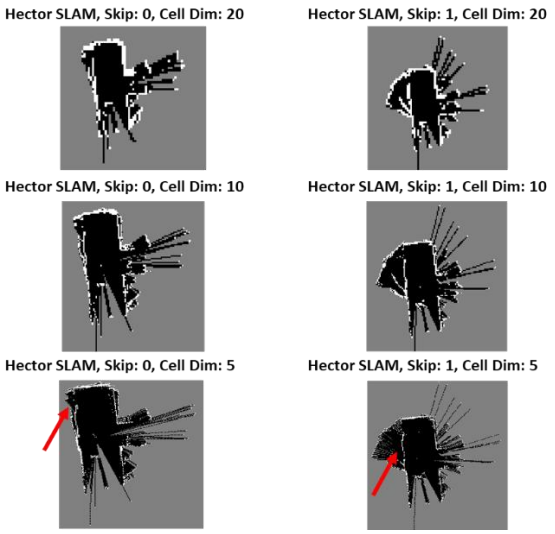

Figure 11. Hector SLAM occupancy grid map solution for different step sizes (no-frame skipped, and one-frame skipped)

	RMSE	Single Scan matching time	Total Time
No-Skipping	0.242 m	0.418 sec	4.599 sec
All Other Trials	stuck in local minima		

Table 3. Hector SLAM Performances (First Experiment)

As shown in Figure 11, even when there is no frame skipped during the solution of multi-level Hector SLAM algorithm, the algorithm failed to accurately construct the map and when at least one frame is skipped to decrease the execution time, it completely failed to converge.

5.1.4 VM-Hector SLAM Performance (First Experiment).

In this experiment, an initialization step is introduced before solving the multi-level Hector SLAM, based on the proposed VM approach. The VM is used to detect the start/end of that harsh manuever and the reading of the gyro during this period is used to estimate the rotation that occurred during the maneuver (106 degrees).
Figure 12 and Table 4 show the ability of the VM initialization step to decrease the execution time and enhance the accuracy of the matching.

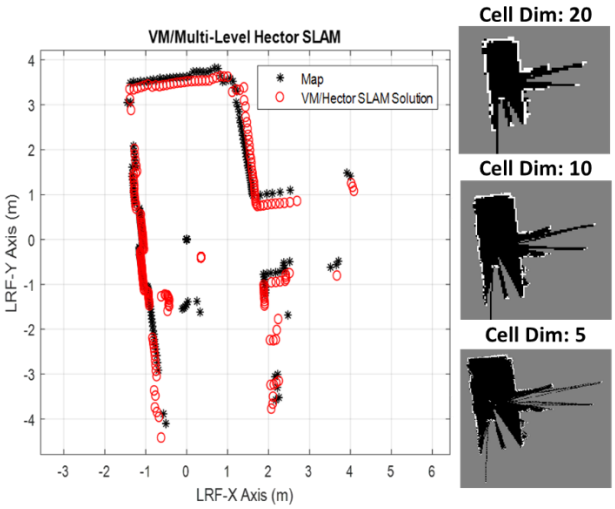

Figure 5. VM/ Multi-level Hector SLAM performance (first experiment)

	RMSE	Total Time
VM/Multi Level	0.203 m	0.552 sec

Table 4. VM/Multi-Level Hector SLAM Performance (First Experiment)

With the aid of the VM initialization step, there is no need to perform multi-level Hector SLAM, only a single Hector SLAM is applicable to achieve a good performance as shown in Figure 13.

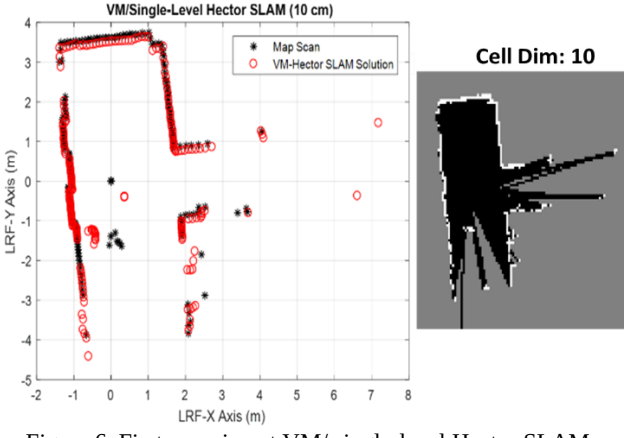

Figure 6. First experiment VM/ single-level Hector SLAM performance (10 cm grid size)

	RMSE	Total Time
VM/ Single Level	0.205 m	0.15 sec

Table 5. VM/Single Level Hector SLAM performance for 10 cm Grid Size (First Experiment).

Figure 13 and Table 5 prove the ability of the initialization VM step to achieve a better performance, even with single-level Hector SLAM, which significantly decreased the time consumption while maintaining the matching accuracy.

5.2 Second Experiment

The aim of the second experiment is to test the proposed approach against the normal ICP when part of the scene is changed because of change in drone height during the harsh rotation. During this experiment, the quadcopter performed a harsh rotation of 120 degrees in 1.4 sec. The total frames taken by LIDAR is 9 frames.

5.2.1 ICP Performance:

Figures 14 and 15 prove that the ICP algorithm stuck in local minima, and it was not able to converge except when there is no frame skipped, but this costs more time and computations as shown in Table 6.

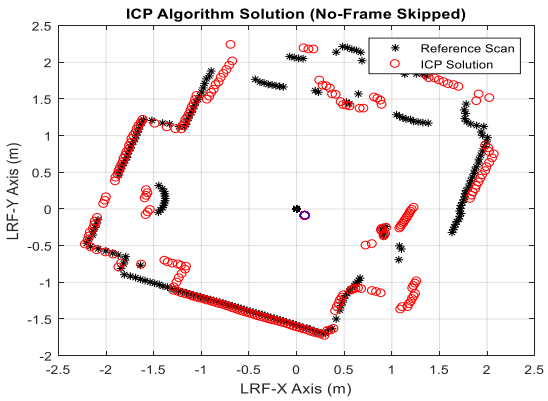

Figure 7. ICP algorithm performance when no frame is skipped (second experiment, no frame skipped).

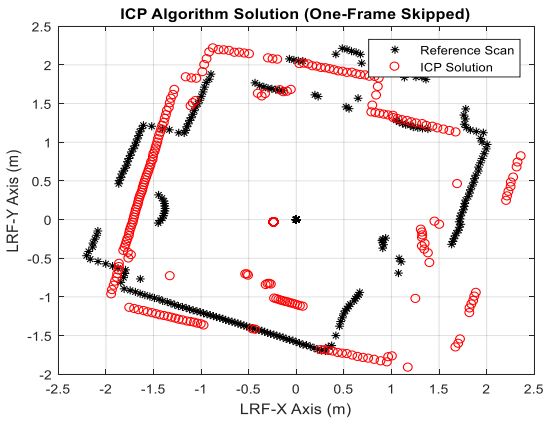

Figure 8. ICP algorithm performance when the step size is one frame skipping (second experiment, one frame skipped).

	RMSE	TIME
No-Frame Skipped	0.145 m	0.1577 sec
With Skipped Scans	Did not Converge	

Table 6. Second Experiment ICP Results

5.2.2 Vehicle Model-ICP

This second experiment was conducted at a different time but the same threshold was utilized to detect the start/end of the maneuver as shown in Figure 16, The estimated rotation using the vehicle model is 116.2 degree. This estimated rotation has been used for coarse alignment, followed by ICP as fine alignment. The result of VM-ICP shown in Figure 17 and Table 7 proves the achieved enhancement after utilizing the proposed approach regarding the execution time.

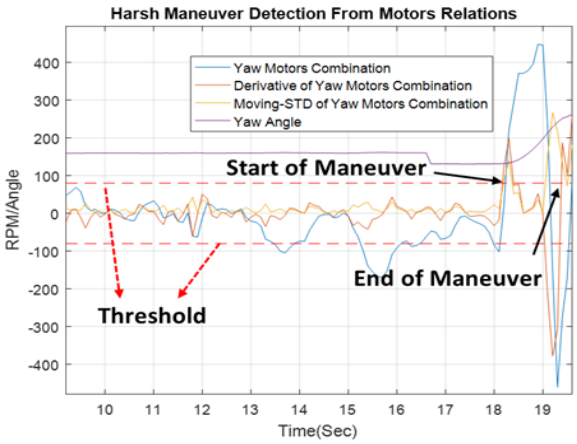

Figure 9. Harsh maneuver detection according to VM motors relations.

	RMSE	TIME
VM-ICP	0.142 m	0.0107 sec

Table 7. VM-ICP Results (Second Experiment)

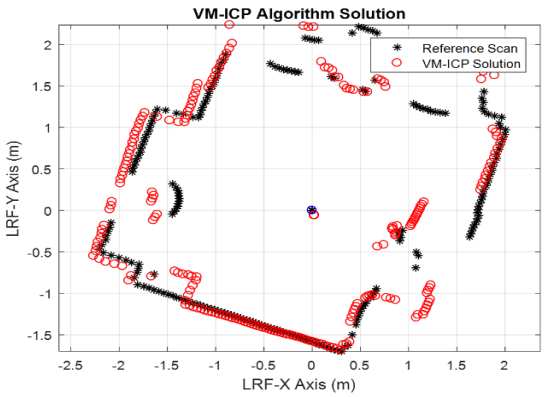

Figure 10. Vehicle-Model ICP performance (second experiment).

5.2.3 Hector SLAM Performance (Second Experiment):

This experiment aims to test the Hector SLAM performance with different numbers of skipped frames regarding the matching performance and the time consumption, even if there is a partial change in the surroundings cause from drone height change.
During this experiment, the Hector SLAM was able only to converge when all the frames were used as shown in Figure 18 and 19, without skipping any frame. The RMSE during this experiment is 0.145 m with mean time 0.3973 sec required to match each scan to the map, with a total time of 3.5 sec to register all the scans during the harsh maneuver.

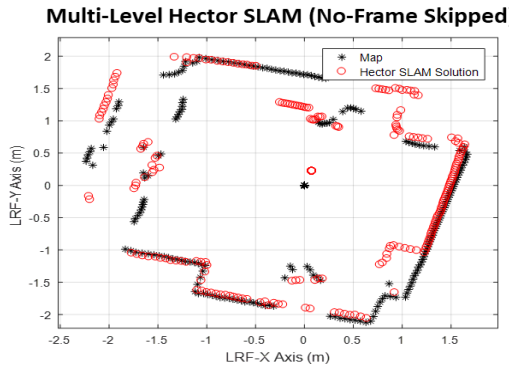

Figure 11. Hector SLAM performance during second experiment with no frame skipped during harsh maneuver

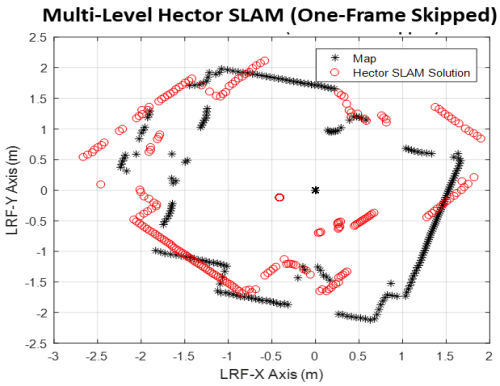

Figure 12. Hector SLAM Performance During Second Experiment with One Frame Skipping During Harsh Maneuver

	RMSE (m)	Mean solution TIME (sec)	Total Time (sec)
No-Frame Skipped	0.1459	0.3973	3.5759
With Skipped Scans	Stuck in a local minima		

Table 8. Hector SLAM Performance During Second Experiment

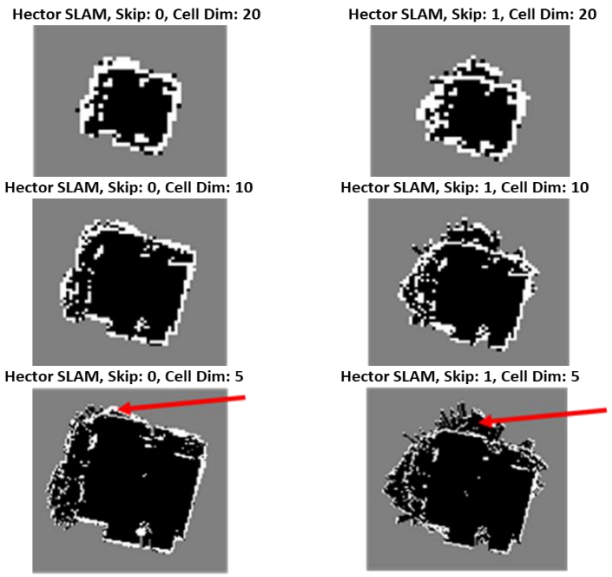

Figure 20. Map Constructed from Hector SLAM During Second Experiment with Different Frames Skipping Steps (No-skipping and One Frame Skipping)

As shown in Figures 18,19, and 20, even when there is no frame skipped during the solution of multi-level Hector SLAM algorithm, the algorithm failed to accurately construct the map and when at least one frame is skipped to decrease the execution time, it completely failed to converge.

5.2.4 Vehicle-Model Hector SLAM Performance (Second Experiment):

This test aims to evaluate the Hector SLAM algorithm performance, and time consumption, after using the VM initialization step, to detect and estimate the amount of rotation during harsh maneuver that may cause partial change in the scene.

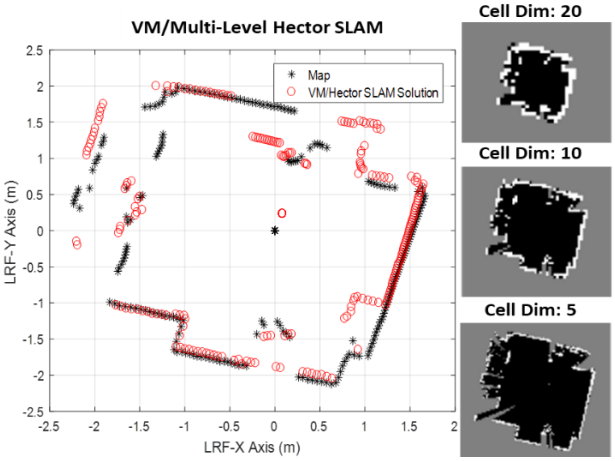

Figure 21. VM- Multi-Level Hector SLAM Performance (Second Experiment)

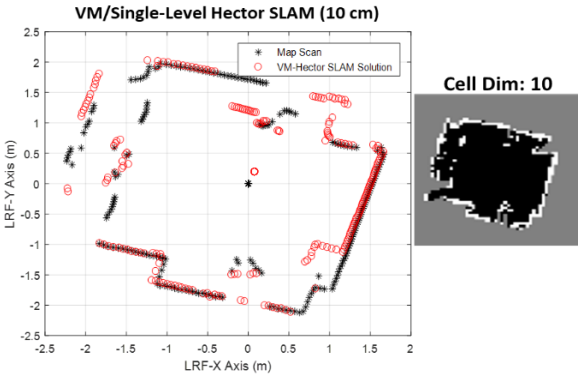

Figure 13. VM- Single-Level Hector SLAM Performance (Second Experiment).

	RMSE	Total Time
VM/Multi Level Hector	0.207 m	0.362 sec
VM/ Single Level Hector	0.156 m	0.112 sec

Table 9. VM/Hector SLAM Performance During Second Experiment (Multilevel and single level)

Figure 21 and 22 and Table 9 shows the ability of the VM-initialization step, to enhance the Hector SLAM algorithm matching performance and the time required for matching, compared with the results of stand-alone Hector SLAM reported in Table 8.

6. CONCLUSIONS

Small/micro UAVs are widely spread in a lot of applications in our daily lives, because of their great maneuverability and small size which made it able to access more challenging places and perform tasks that may expose human lives to danger while saving time, cost, and effort. The small size and cost of these UAVs also impose some limitations on the onboard sensors

utilized for navigation regarding size, cost, and power consumption. These small UAVs navigation mainly depends on the fusion between the GNSS and IMU, but during GNSS signal unavailability, the UAVs suffers from massive deterioration in the navigation states estimation, because of the rapidly accumulated errors of the low-cost MEMS-based IMU. In the indoor environment, the GNSS is not available, so other aiding sensors are typically required to localize the UAV. Scan matching algorithms such as ICP/Hector SLAM are among the mainly used approaches in unstructured indoor environments with the aid of LIDAR, to estimate the rotation and translation occurred from scan to scan. The iterative nature of such approaches costs the system more computations and time. Harsh maneuvers can significantly affect the performance of such algorithms as their matching concept is dramatically affected by the mismatches of the initial iterations under harsh maneuvers. Such behavior increases the required time in the attempt of finding an accurate transformation and also increases the chances of falling in local minima or divergence. The proposed approach utilizes the vehicle model to detect such harsh maneuvers and then estimates its transformation to serve as an initial estimate (coarse alignment). This coarse step is followed by normal ICP/Hector SLAM algorithm to serve as fine alignment. This proposed approach utilizing the vehicle model does not require extra sensors as it employs the speeds of the UAV motors to detect and estimate this harsh movement.

Two experiments have been conducted, that included harsh maneuver that reached 2.5 rad/s, to evaluate the performance of the VM-ICP approach against normal ICP, and VM-Hector against normal multi-level Hector SLAM algorithm. These experiments show that the VM-ICP solution in the first experiment reached RMSE of 0.17 m in 0.0023 sec, while the ICP RMSE reached 0.81 m in 0.014 sec or 0.21 m in 0.096 sec according to the number of frames skipped during solution. The VM-ICP in the second experiment exhibited an RMSE of 0.14 m in 0.01 sec, while ICP solutions failed to converge except when the algorithm used all the scans from LIDAR with RMSE 0.145 m in 0.15 sec. The VM initialization step also caused great enhancement when combined with Hector SLAM, as it alleviates the need to perform multi-level Hector SLAM. The result of the VM Multi-Level Hector slam and VM Single-Level Hector SLAM in the first experiment where 0.203 m RMSE in 0.552 sec, and 0.205 m RMSE in 0.15 sec respectively, compared to normal Hector SLAM which achieved 0.242 m RMSE in 4.5 sec, while the Hector SLAM in the second experiment achieved 0.145 m RMSE in 3.5 sec, compared to Single-Level and Multi-Level VM Hector SLAM, that achieved 0.207 m RMSE in 0.362 sec and 0.156 m RMSE in 0.112 sec respectively. These experiments prove the ability of the proposed VM initialization step, to improve the performance of iterative scan matching algorithms such ICP and Hector SLAM regarding both the matching RMSE and the time consumption. Moreover, this proposed approach doesn't require any features to work, so it is able to enhance scan matching performance in a completely unstructured environment.

ACKNOWLEDGMENT

This work was supported by Dr. Naser El-Sheimy research funds from NSERC and Canada Research Chairs programs. Thanks also go to the funding of the first author by the Egyptian government.

REFERENCES

Cadena, C., Carlone, L., Latif, Y., Scaramuzza, D., Neira, J., Reid, I., and Leonard, J., 2016. Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age, IEEE Trans. Robot., vol. 32, no. 6, pp. 1309–1332, Dec.

Censi,A. , 2008. An ICP Variant Using A Point-To-Line Metric, In Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on, pp. 19–25.

Chugo, D., Yokota, S., Matsushima, S., and Takase, K. 2010. Camera-Based Indoor Navigation for Service Robots, in Proceedings of SICE Annual Conference 2010, pp. 1008–1013.

Diosi, A., and Kleeman, L. 2007. Fast Laser Scan Matching Using Polar Coordinates, Int. J. Robot. Res., vol. 26, no. 10, pp. 1125–1153.

Gao, Y., Liu, S., Atia, M., and Noureldin, A., 2015. INS/GPS/LiDAR Integrated Navigation System for Urban and Indoor Environments Using Hybrid Scan Matching Algorithm, *Sensors*, vol. 15, no. 9, pp. 23286–23302, Sep.

Grzonka, S., Grisetti, G., and Burgard, W., 2009. Towards A Navigation System for Autonomous Indoor Flying, in 2009 IEEE International Conference on Robotics and Automation, pp. 2878–2883.

Holm, S., and Nilsen, C., 2010. Robust Ultrasonic Indoor Positioning Using Transmitter Arrays, in 2010 International Conference on Indoor Positioning and Indoor Navigation, 2010, pp. 1–5.

Kohlbrecher, S., Meyer, J., Graber, T., Petersen, K., Klingauf, U., and von Stryk, O., 2013. Hector Open Source Modules for Autonomous Mapping And Navigation With Rescue Robots, in Robot Soccer World Cup, pp. 624–631.

Mohamed, H., Moussa, A., Elhabiby, M., El-Sheimy, N. and Sesay, A. 2016. Improved Real-Time Scan Matching Using Corner Features," ISPRS - Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci., vol. XLI-B5, pp. 533–539, Jun.

Mohamed, H., Moussa, A., Elhabiby, M., El-Sheimy, N. and Sesay, A.2017. A Novel Real-Time Reference Key Frame Scan Matching Method, Sensors, vol. 17, no. 5, May.

Mohamed, H., Moussa, A., Elhabiby, M., El-Sheimy, N. and Sesay, A. 2017. Corner Features Aided Indoor Slam For Unmanned Vehicles, in The 10th International Symposium on Mobile Mapping Technology, Cairo, Egypt.

R. Mahony, V. Kumar, and P. Corke, "Multirotor Aerial Vehicles: Modeling, Estimation, and Control of Quadrotor," IEEE Robot. Autom. Mag., vol. 19, no. 3, pp. 20–32, Sep. 2012.

Schnipke, E., Nemati, A., Sarim, M., Heshemi, M., Redifling, S., Jeffers, W.. Meiring J., Sampathkumar, P. and Kumar, M. Autonomous Navigation of UAV through GPS-Denied Indoor Environment with Obstacles, in AIAA Infotech @ Aerospace, American Institute of Aeronautics and Astronautics.

Shu, L., Xu, H., and Huang, M. 213. High-Speed and Accurate Laser Scan Matching Using Classified Features, in 2013 IEEE International Symposium on Robotic and Sensors Environments (ROSE), pp. 61–66.

Ulaş C., and Temeltaş, H. 2011. A Fast and Robust Scan Matching Algorithm Based On Feature Dependent Sampling, in 2011 9th IEEE International Conference on Control and Automation (ICCA), pp. 124–129.

Wang, F., Wang, K., Lai, S., Phang, S., Chen, B. and Lee, T. 2014. An Efficient UAV Navigation Solution for Confined but Partially Known Indoor Environments, in 11th IEEE International Conference on Control Automation (ICCA), pp. 1351–1356.

Zhang, Z. 1994. Iterative Point Matching for Registration of Free-Form Curves and Surfaces, Int. J. Comput. Vis., vol. 13, no. 2, pp. 119–152.