2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

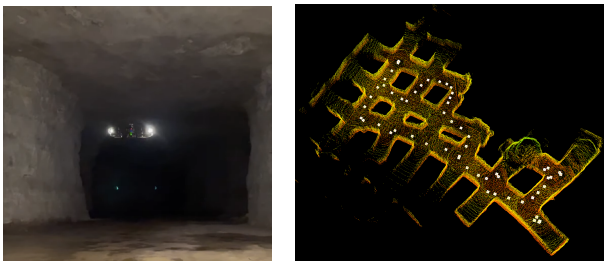
Adaptive Keyframe Generation based LiDAR Inertial Odometry for Complex Underground Environments

Boseong Kim[1], Chanyoung Jung[1], D. Hyunchul Shim[1] and Ali-akbar Agha-mohammadi[2]

***Abstract*— In this paper, we present a LiDAR Inertial Odometry (LIO) algorithm utilizing adaptive keyframe generation which achieves fast and accurate state estimation for aerial and ground robots. It is known that keyframe generation significantly affects the performance of Simultaneous Localization and Mapping (SLAM) algorithms. Unlike existing SLAM algorithms that generate keyframes based on fixed conditions, we propose to use adaptive keyframe generation conditions considering characteristics of surrounding environment using real-time LiDAR scans. When a keyframe is generated, the keyframe and the corresponding LiDAR measurements are stored in our novel data structure designed for efficient submap generation. The *scan to sub-map* matching module then uses the Generalized Iterative Closest Point (GICP) algorithm to adjust estimated states at a global scale, producing more accurate and globally consistent state estimation results even in large-scale underground environments. Experimental results from diverse types of underground environments show that the proposed method outperforms the existing state-of-the-art LIO algorithms in various metrics such as computational speed, CPU usage, and accuracy.**

I. INTRODUCTION

State estimation is the most essential element for the operation of unmanned systems such as drones and mobile robots. Recently, State estimation algorithms using LiDAR, cameras and IMU are being actively studied. Among them, LiDAR-based algorithms have the advantage of accurate state estimation performance and are very commonly used for autonomous systems. Also, if tightly coupled with an IMU, state estimation accuracy can be greatly improved even in aggressive motion by compensating the relative position and orientation change within a short time. Due to its small and lightweight hardware characteristics, vision-based state estimation algorithms [1]–[4] using a stereo or monocular camera have a great advantage in the field of unmanned aerial vehicles. However, even with these advantages, the vision-based state estimation algorithms have a sensitive problem of failing to extract feature points in visually degraded environments such as very dark, foggy and dusty areas. Therefore, in this paper, we focus on LiDAR-based state estimation algorithms for reliable localization even in perceptually degraded environments such as in underground environments.



(a) (b)

Fig. 1: We (Team CoSTAR) conducted an autonomous flight experiment inside a limestone cave in Wilmore, KY to participate in the DARPA subterranean challenges final event. (a) Proxima, the customized drone platform of Team CoSTAR. (b) The map generated by Proxima using the proposed method while autonomously flying. The white dots indicate keyframes for map generation and the generation conditions are changed according to the wideness of the surrounding environment.

It has been demonstrated that the LiDAR-based state estimation algorithms are more robust than the vision-based algorithms in visually degraded environments by using accurate depth information. For example, the most widely used LOAM [5] is a real-time mapping system that has advantages such as low drift and low computational cost by using a feature extraction technique that extracts plane and edge features. Based on the feature extraction method of LOAM, F-LOAM [6], which uses a non-iterative two-stage distortion compensation method to reduce the computational cost, has been proposed. In addition, LeGO-LOAM [7] with ground-optimized characteristics and LIO-SAM [8], which is tightly coupled with an IMU, have been proposed. However, depending on the surrounding environment, these feature extraction methods do not use sufficient LiDAR measurements and thus state estimation performance is degraded. Also in case of long term state estimation, the state estimation accuracy is not guaranteed because local relative pose estimation drifts accumulate over time causing large inconsistency at a global scale. If this feature extraction method is not used, the number of LiDAR points becomes too large in large environments and the computational cost required to estimate the relative position between two consecutive LiDAR frames can be high. This causes LiDAR frame drops which means that the overall localization performance deteriorates.

*This research was financially supported by the Institute of Civil Military Technology Cooperation funded by the Defense Acquisition Program Administration and Ministry of Trade, Industry and Energy of Korean government under grant No. UM22206RD2.

[1]Boseong Kim, Chanyoung Jung and D. Hyunchul Shim are with the Department of Electrical Engineering, Korea Advanced Institute of Science and Technology (KAIST), Yuseong-gu, Daejeon 34141, Republic of Korea (email: {brian.kim, hy910915, hcshim}@kaist.ac.kr).

[2]Ali-akbar Agha-mohammadi is with the NASA Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91125 USA (e-mail: aliagha@jpl.nasa.gov).

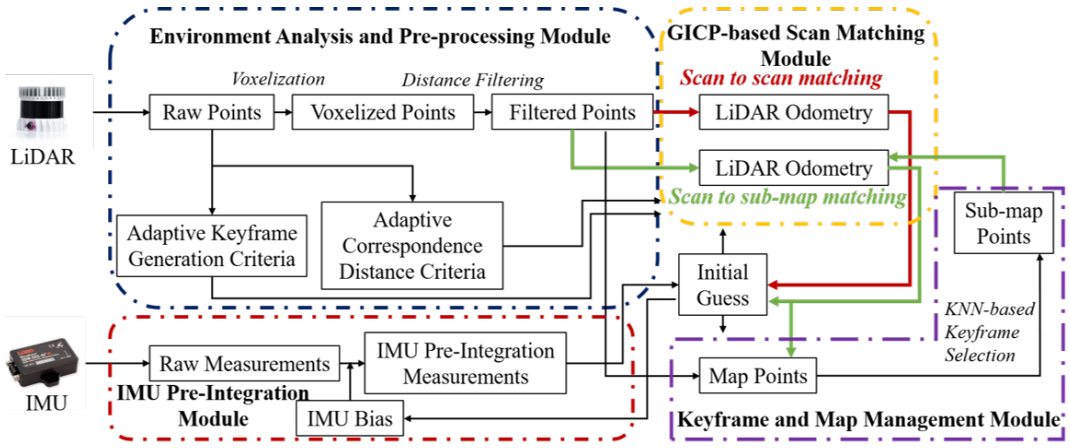


Fig. 2: The system configuration of the proposed method. The system consists of four modules: IMU pre-integration, Environment analysis and pre-processing, GICP-based scan matching and Keyframe and map management module. The red and green lines in the scan matching module represent *scan to scan* matching and *scan to sub-map* matching sequences respectively.

In this paper, we propose an adaptive keyframe based LiDAR Inertial Odometry (LIO) algorithm that can balance computational cost and accuracy through surrounding environment analysis. Specifically, the proposed method can reduce the computational cost by generating sparse keyframes in large environments and increase accuracy by generating dense keyframes in narrow environments. In comparison with benchmark algorithms, the proposed method shows the highest accuracy not only in general environments but also in extreme environments.

The main contributions of this paper are as follows:

- We proposed a hierarchical hybrid LO/LIO architecture that balances computational constraints and accuracy using adaptive keyframe generation from surrounding environment analysis.
- Our novel keyframe management module enables fast sub-map generation for *scan to sub-map* matching, enabling accurate localization in large-scale environments.
- The most important contribution of this paper is that we deployed the proposed method to real robots in perceptually degraded environments and extensively validated the performance of the proposed method.

II. RELATED WORKS

Recently, LiDAR-based state estimation algorithms have been actively proposed for reliable unmanned system operation. In addition, many LiDAR Inertial Odometry (LIO) methods tightly coupled with an IMU have been proposed by using the IMU pre-integration method [9]. Their performance has been verified in various environments according to LiDAR point pre-processing and scan matching methods.

One of the most widely used LiDAR measurements pre-processing method is the feature extraction method. These methods are based on the feature extraction method of LOAM which evaluates the roughness in local regions. Extracted edge and planar points are used to compute the relative transformation between two consecutive LiDAR frames through Levenberg-Marquardt (LM) optimization [10]. LIO-SAM [8] is a tightly coupled LIO algorithm with an IMU that improves the estimation speed and accuracy based on the sub-keyframes generation using a sliding window. In addition, global scale position can be compensated for by using additional sensors such as GPS. Also, LIOM [11] and LINS [12], which are tightly coupled LIOs, use the joint-optimization method and an iterated error-state Kalman filter (ESKF) respectively to improve accuracy and real-time processing.

Rather than just the feature points, the point-based scan matching method using an entire set of LiDAR measurements has the advantage of improving the state estimation accuracy by fully utilizing the spatial information of the environment. The Iterative Closest Point (ICP) [13] [14] method is the most widely used scan matching algorithm; Generalized ICP (GICP) [15] and Voxelized GICP (VGICP) [16], which improve accuracy and computational performance, have also been proposed. HDL-graph-SLAM [17] is a real-time high definition SLAM algorithm involving multiple point-based scan matching methods such as NDT, ICP, and GICP. LiTAMIN [18] proposed ICP-based scan matching using LiDAR data applied with normal distribution to achieve fast and stable performance.

However, these conventional LIO methods do not guarantee the reliability of long-term state estimation because there is no position compensation at the global scale except for loop closure. In addition, the state estimation performance is degraded in complex and large-scale environments such as multi-floor and underground environments due to the fixed keyframe generation conditions.

III. METHOD

A. System Overview

The robot state $\mathbf{X}$ can be expressed as follows:

$$\mathbf{X} = \begin{bmatrix} R^T, & p^T, & v^T, & b_a^T, & b_\omega^T \end{bmatrix}^T, \tag{1}$$

where $R \in SO(3)$, $p \in \mathbb{R}^3$, $v \in \mathbb{R}^3$, $b_a \in \mathbb{R}^3$ and $b_\omega \in \mathbb{R}^3$ are the rotation matrix, position vector, velocity vector, accelerometer and gyroscope bias vector of the IMU, respectively. Note that $R^{\mathbf{WB}}$ is the rotation matrix from robot body frame **B** to world frame **W**. In this paper, for convenience, we assume that the IMU frame and **B** are the same. Also the transformation matrix $\mathbf{T} \in SE(3)$ from **B** to **W** can be represented as $\mathbf{T} = \begin{bmatrix} R & p \end{bmatrix}$. An overview of the proposed method is shown in Fig. 2.

B. IMU Pre-Integration

The IMU pre-integration module uses raw IMU measurements to calculate the relative position, velocity and rotation changes between the previous IMU frame and the current IMU frame. We can define the IMU raw measurement model as follows:

$$\hat{\omega}_t = \omega_t + b_t^\omega + n_t^\omega \tag{2}$$

$$\hat{a}_t = R_t^{\mathbf{BW}}(a_t - \mathrm{g}) + b_t^a + n_t^a, \tag{3}$$

where $\hat{\omega}_t$ and $\hat{a}_t$ are raw IMU measurements in **B** at time t and g is the gravity vector in **W**. b_t^ω and b_t^a are biases of the gyroscope and accelerometer respectively that can be modeled by random walk. Also, n_t^ω and n_t^a are modeled as Gaussian white noise of the gyroscope and accelerometer. After receiving the IMU measurements we can obtain the relative motion between the i and j frames by applying the IMU pre-integration technique proposed in [9]. The IMU pre-integration measurements Δp_{ij}, Δv_{ij} and Δq_{ij} can be computed as follows:

$$\Delta p_{ij} = \sum_{k=i}^{j-1} \Big[\Delta v_{ik} \Delta t + \frac{1}{2} R(\Delta q_{ik})(\hat{a}_k - b_k^a)\Delta t^2 \Big] \tag{4}$$

$$\Delta v_{ij} = \sum_{k=i}^{j-1} R(\Delta q_{ik})(\hat{a}_k - b_k^a)\Delta t \tag{5}$$

$$\Delta q_{ij} = q_i^{-1} \otimes q_j = \prod_{k=i}^{j-1} \begin{bmatrix} \frac{1}{2}\Delta t(\hat{\omega}_k - b_k^\omega) \\ 1 \end{bmatrix}, \tag{6}$$

where $R(\cdot)$, $\otimes$ and Δt are the converting symbol from quaternion to rotation matrix, the quaternion multiplication symbol and the time interval between two consecutive IMU frames, respectively. We suggest that readers refer to [9] for a detailed explanation of (4), (5) and (6).

The IMU pre-integration measurements were used as an initial guess when performing GICP-based scan matching to compute relative transformation between two consecutive LiDAR frames in the scan matching module. The IMU bias was optimized through non-linear optimization with LiDAR odometry obtained from the scan matching module.

Note that the proposed method is a hybrid LO/LIO algorithm. If there is no IMU input, this module is not used and the initial guess is considered as previous state.

C. Environment Analysis and Pre-Processing

After obtaining LiDAR measurements we first define the wideness of the surrounding environment. In wide environments, even when LiDAR points downsampling was performed using large voxelization parameters, sufficient geometric features were maintained. So even if the keyframe generation interval was large, accurate state estimation using sparse points was possible. On the other hand, in narrow environments, since the environment can change rapidly due to blind spots, a large voxelization parameter can distort geometrical features. Therefore the keyframe generation interval must be dense. We also changed the correspondence distance parameter used for *scan to sub-map* matching according to the wideness of the surrounding environment. For wide areas the maximum correspondence distance parameter was set to a large value because the keyframe generation interval was very large and the common geometrical features were sufficient for a long time. On the other hand, in narrow areas such as stairs and narrow corners, the maximum correspondence distance parameter was set to a small value because

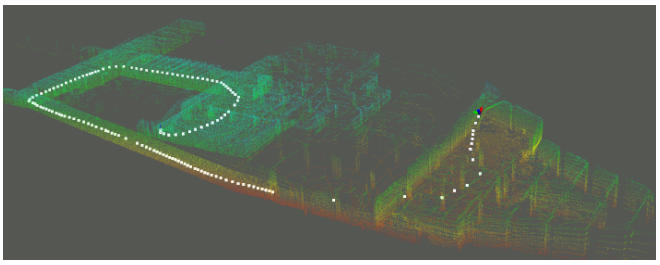


Fig. 3: Adaptive keyframe generation. The white dots represent the generated keyframes. We define the wideness of the environment according to $N_t^{V_{\text{env}}}$. d_{key} and d_{corr} also depend on $N_t^{V_{\text{env}}}$.

Algorithm 1 Environment analysis and pre-processing

Input: Current LiDAR scan $\hat{\mathbf{M}}_t$, voxel size V_{env}(10m), maximum number of points $N_{\text{max}}^{V_{\text{env}}}$, maximum voxelization size $V_{\text{scan}}^{\text{max}}$, maximum keyframe distance $d_{\text{key}}^{\text{max}}$, maximum correspondence distance criteria $d_{\text{corr}}^{\text{max}}$, exponential decay value α, β, γ and μ

Output: Voxelization parameter for scan matching V_{scan}, Keyframe generation criteria d_{key}, Maximum correspondence distance d_{corr} and Filtered scan $\mathbf{F}_t^{V_{\text{scan}}}$.

$'\mathbf{M}_t \leftarrow$ Outlier filtering($\hat{\mathbf{M}}_t$) ▷ Initialization
$'\mathbf{M}_t^{V_{\text{env}}} \leftarrow$ Voxelization($'\mathbf{M}_t$) with V_{env}
$N_t^{V_{\text{env}}} \leftarrow$ Number of points($'\mathbf{M}_t^{V_{\text{env}}}$)
if $N_t^{V_{\text{env}}} \geq N_{\text{max}}^{V_{\text{env}}}$ **then** ▷ Super wide
  $V_{\text{scan}} = V_{\text{scan}}^{\text{max}}, d_{\text{key}} = d_{\text{key}}^{\text{max}}$ and $d_{\text{corr}} = d_{\text{corr}}^{\text{max}}$
else if $N_t^{V_{\text{env}}} \geq \alpha N_{\text{max}}^{V_{\text{env}}}$ **then** ▷ Wide
  $V_{\text{scan}} = \beta V_{\text{scan}}^{\text{max}}, d_{\text{key}} = \gamma d_{\text{key}}^{\text{max}}$ and $d_{\text{corr}} = \mu d_{\text{corr}}^{\text{max}}$
else if $N_t^{V_{\text{env}}} \geq \alpha^2 N_{\text{max}}^{V_{\text{env}}}$ **then** ▷ Normal
  $V_{\text{scan}} = \beta^2 V_{\text{scan}}^{\text{max}}, d_{\text{key}} = \gamma^2 d_{\text{key}}^{\text{max}}$ and $d_{\text{corr}} = \mu^2 d_{\text{corr}}^{\text{max}}$
else if $N_t^{V_{\text{env}}} \geq \alpha^3 N_{\text{max}}^{V_{\text{env}}}$ **then** ▷ Narrow
  $V_{\text{scan}} = \beta^3 V_{\text{scan}}^{\text{max}}, d_{\text{key}} = \gamma^3 d_{\text{key}}^{\text{max}}$ and $d_{\text{corr}} = \mu^3 d_{\text{corr}}^{\text{max}}$
else ▷ Super narrow
  $V_{\text{scan}} = \beta^4 V_{\text{scan}}^{\text{max}}, d_{\text{key}} = \gamma^4 d_{\text{key}}^{\text{max}}$ and $d_{\text{corr}} = \mu^4 d_{\text{corr}}^{\text{max}}$
end if
$\hat{\mathbf{F}}_t^{V_{\text{scan}}} \leftarrow$ Radius filtering($'\mathbf{M}_t^{V_{\text{scan}}}$) with radius R_{robot}

there were cases in which common geometrical features were not sufficient due to sudden environmental changes. Therefore, as described in Algorithm 1, we defined the parameters to be used for scan matching module according to the surrounding environment and this enabled the proposed method to perform global scale position compensation in large areas and local scale position compensation in narrow areas. The keyframes generated according to the result of Algorithm 1 are shown in Fig. 3.

D. GICP-based Scan Matching

The output of Algorithm 1 was transmitted to the scan matching module to compute the relative transformation of two consecutive LiDAR frames. To use sufficient spatial information we used point-based scan matching rather than feature-based scan matching; the GICP [15] method was used for scan matching.

Two consecutive LiDAR measurements $\hat{\mathbf{Z}}_{t-1} = \{\hat{Z}_{i,t-1}\}$ and $\hat{\mathbf{Z}}_t = \{\hat{Z}_{i,t}\}$ can be expressed by the following probabilistic model:

$$Z_{i,t-1} \sim \mathcal{N}(\hat{Z}_{i,t-1}, C_{i,t-1}), \quad Z_{i,t} \sim \mathcal{N}(\hat{Z}_{i,t}, C_{i,t}), \quad (7)$$

where $C_{i,t-1}$ and $C_{i,t}$ are covariance matrices. Assuming that there is no noise and there are only perfect correspondences, the relative transformation $\mathbf{T}^*$ between $\hat{Z}_{i,t-1}$ and $\hat{Z}_{i,t}$ satisfies $\hat{Z}_{i,t} = \mathbf{T}^* \hat{Z}_{i,t-1}$. We can define the Euclidean distance $d_{i,t}^{\mathbf{T}}$ with arbitrary transform $\mathbf{T}$ as follow:

$$d_{i,t}^{\mathbf{T}} = \hat{Z}_{i,t} - \mathbf{T}\hat{Z}_{i,t-1}, \quad (8)$$

$$d_{i,t}^{\mathbf{T}^*} \sim \mathcal{N}(0, C_{i,t} + (\mathbf{T}^*) C_{i,t-1} (\mathbf{T}^*)^T). \quad (9)$$

Now we can compute the relative transformations for two consecutive LiDAR frames from (9) through iterative calculations using the maximum likelihood estimation (MLE) as follows:

$$\mathbf{T} = \arg\min_{\mathbf{T}} \sum_i (d_{i,t}^{\mathbf{T}})^T (C_{i,t} + \mathbf{T} C_{i,t-1} \mathbf{T}^T)^{-1} (d_{i,t}^{\mathbf{T}}). \quad (10)$$

As mentioned previously, the scan matching module consists of *scan to scan* matching and *scan to sub-map* matching. *Scan to scan* matching computes the relative transformation $\mathbf{T}$ between two consecutive LiDAR frames; *scan to sub-map* matching is performed when the keyframe is generated by satisfying d_{key} defined in Algorithm 1. The state estimated through the last *scan to sub-map* matching is set as the reference state $\mathbf{T}_L^*$ in W and the state between consecutive keyframes is estimated by accumulating the results of *scan to scan* matching. From (10), when the relative transformation between the last keyframe and time t is $\Delta\mathbf{T}_t$, the robot state $\mathbf{T}_t$ at time t in $\mathbf{W}$ can be expressed as follows:

$$\mathbf{T}_t = \mathbf{T}_L^* \Delta\mathbf{T}_t \quad (11)$$

$$\Delta\mathbf{T}_t = \arg\min_{\mathbf{T}} \sum_i ({}^{\text{s}}d_{i,t}^{\mathbf{T}})^T (C_{i,t} + \mathbf{T} C_{i,L} \mathbf{T}^T)^{-1} ({}^{\text{s}}d_{i,t}^{\mathbf{T}}) \quad (12)$$

$$\mathbf{T}_L^* = \arg\min_{\mathbf{T}} \sum_i ({}^{\text{m}}d_{i,t}^{\mathbf{T}})^T (C_{i,t} + \mathbf{T} C_{i,M} \mathbf{T}^T)^{-1} ({}^{\text{m}}d_{i,t}^{\mathbf{T}}), \quad (13)$$

Algorithm 2 kNN-based Sub-map generation

Input: $\hat{\mathbf{M}}_t$, $\Delta\mathbf{T}_t$, d_{key}, $\mathbf{T}_t$ and $\mathbf{T}_L^*$
Output: All keyframes $\mathbf{K} = \{K_i\}$, Entire map points $\hat{\mathbf{M}}_A = \{\hat{\mathbf{M}}_{i,A}\}$, k-nearest keyframes $\mathbf{K}_k$ and KNN-based sub-map $\hat{\mathbf{M}}_M$

$\mathbf{T}_L^*, \Delta\mathbf{T}_t \leftarrow I \quad i = 0$ ▷ Initialization
$K_0 = p(\mathbf{T}_L^*)$ and $\hat{\mathbf{M}}_{0,A} = \hat{\mathbf{M}}_t \mathbf{T}_L^*$ ▷ Initialization
if $\|p(\Delta\mathbf{T}_t)\| \geq d_{\text{key}}$ **then**
  KDtree.Build($\mathbf{K}$)
  KDtree.Search($\mathbf{T}_t, k$)
  $\{k_0, k_1, \cdots, k_{k-1}\} \leftarrow k$**NN**.ExtractIndex
  $\mathbf{K}_k \leftarrow \{K_{k_0}, K_{k_1}, \cdots, K_{k_{k-1}}\}$
  $\hat{\mathbf{M}}_M \leftarrow \{\hat{\mathbf{M}}_{k_0,A}, \hat{\mathbf{M}}_{k_1,A}, \cdots, \hat{\mathbf{M}}_{k_{k-1},A}\}$
  GICP.BuildTarget($\hat{\mathbf{F}}_M^{V_{\text{scan}}}$, C_M) ▷ Section III-D
  $\mathbf{T}_L^* \leftarrow$ **GICP**.GetTransform($\mathbf{T}_t, d_{\text{corr}}$)
  $RegistMap = 1$
  for $j = 0$ to $i - 1$ **do**
    if $\|p(\mathbf{T}_L^*) - K_j\| \leq d_{\text{key}}$ **then**
      $RegistMap = 0$ ▷ To prevent overlap
    end if
  end for
  if $RegistMap$ **then**
    $\mathbf{K}$.PushBack $(p(\mathbf{T}_L^*))$
    $\hat{\mathbf{M}}_A$.PushBack $(\hat{\mathbf{M}}_t \mathbf{T}_L^*)$
  end if
  $i \leftarrow i + 1$
end if
$p(\mathbf{T})$ *is the position vector* p *in* $\mathbf{T} = \begin{bmatrix} R & | & p \end{bmatrix}$

where $C_{i,L}$ and $C_{i,M}$ are covariance matrices of LiDAR measurements of the last keyframe and sub-map points, respectively. The error models considering the d_{corr} in Section III-C for *scan to scan* matching $({}^{\text{s}}d_{i,t}^{\mathbf{T}})$ and *scan to sub-map* matching $({}^{\text{m}}d_{i,t}^{\mathbf{T}})$ can be expressed from (8) and Algorithm 1 as follows:

$${}^{\text{s}}d_{i,t}^{\mathbf{T}} = \hat{\mathbf{F}}_{i,t}^{V_{\text{scan}}} - \mathbf{T}\hat{\mathbf{F}}_{i,L}^{V_{\text{scan}}} \quad (14)$$

$${}^{\text{m}}d_{i,t}^{\mathbf{T}} = \hat{\mathbf{F}}_{i,t}^{V_{\text{scan}}} - \mathbf{T}\hat{\mathbf{F}}_{i,M}^{V_{\text{scan}}}, \quad (15)$$

where $\hat{\mathbf{F}}_{i,L}^{V_{\text{scan}}}$ and $\hat{\mathbf{F}}_{i,M}^{V_{\text{scan}}}$ are the filtered points of the last keyframe's LiDAR measurements and sub-map points by Algorithm 1. The details of sub-map points will be introduced in Section III-E. Finally, when $\Delta\mathbf{T}_t$ was obtained and an IMU is available, the IMU bias in Section III-B was optimized using $\mathbf{T}_t$.

E. Keyframe and Map Data Management

As mentioned in Section III-D, *scan to sub-map* matching is an important factor to improve the state estimation performance of the proposed method. In this module, we propose a kNN-based sub-map generation method that can employ sufficient spatial information while maintaining lower computational cost than the radius-based filtering method as shown in Figs. 4 and 5. For fast $\hat{\mathbf{M}}_M$ generation, we propose the kNN-based sub-map generation method described in

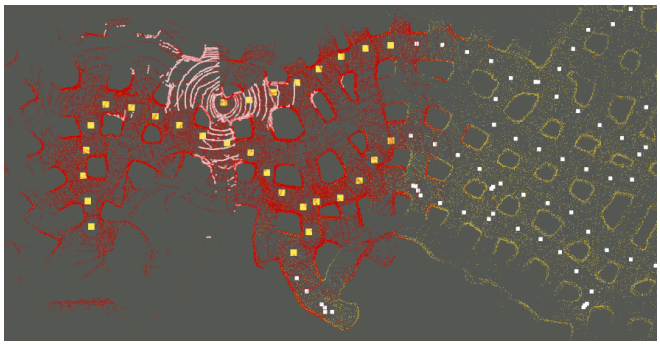


Fig. 4: *Scan to sub-map* matching via kNN-based sub-map generation. White squares, yellow squares, yellow map, red map and white scan represent $\mathbf{K}$, $\mathbf{K}_k$, $\hat{\mathbf{M}}_A$, $\hat{\mathbf{M}}_M$ and $\hat{\mathbf{M}}_t \mathbf{T}_L^*$, respectively, in Algorithm 2.

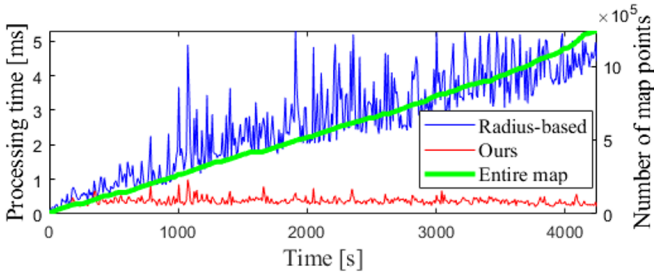


Fig. 5: Comparison of processing time for sub-map generation. As the overall map size increases over time, the computational cost of the radius-based (80m) method increases while that of the proposed kNN-based ($k = 30$) method stays low.

TABLE I: Datasets details

Dataset	Length [m]	LiDAR	Platform	Type
LA subway	656.41	Ouster 32ch	Proxima	Long tunnel
Kentucky cave	632.21	Ouster 32ch	Proxima	Limestone cave
Multi-floor	851.03	Velodyne 16ch	Boston Dynamics Spot	Extreme multi-floor
Mega Cavern	9118.11	Velodyne 16ch	Ground vehicle	Huge cave
DARPA SubT Final Event	522.73	Ouster 32ch	Proxima	Narrow and long corridor
Wells cave	368.95	Ouster 32ch	Proxima	Narrow cave

Algorithm 2. In Algorithm 2, when the keyframe generation interval defined in Algorithm 1 is satisfied, k keyframes are extracted using the kNN algorithm and a sub-map is generated at the same time. Finally, scan matching between the sub-map and the current filtered scan is performed for position compensation.

IV. Experiments

A. Experiment Setup

Using multiple datasets, we compared our proposed method with both LO and LIO algorithms such as LeGO LOAM [7], HDL-graph-SLAM [17], LINS [12] and LIO-SAM [8]. In all experiments we used the same values of $R_{\text{robot}} = 0.5$m, $N_{\text{max}}^{V_{\text{env}}} = 100$, $V_{\text{scan}}^{\text{max}} = 1.0$, $d_{\text{key}}^{\text{max}} = 20$m, $d_{\text{corr}}^{\text{max}} = 5$m, $\alpha = 0.6$, $\beta = 0.7$, $\gamma = 0.5$, $\mu = 0.5$ and $k = 30$ in Algorithms 1 and 2. FLANN [19] was used for the kNN

TABLE II: LiDAR-only Odometry (LO) comparison results in general environment datasets. (Red: Best, Blue: Second best)

Dataset	Method	e2e Error [m]	Computational time [ms]	CPU [%]
LA subway	**Proposed**	0.041	16.33	9.87
	HDL-NDT [17]	4.255	26.31	13.24
	HDL-Fast GICP [17]	6.128	21.34	10.18
	LeGO-LOAM [7]	Fail	10.04	8.37
Kentucky cave	**Proposed**	0.044	18.40	12.74
	HDL-NDT [17]	2.304	30.32	18.12
	HDL-Fast GICP [17]	1.123	24.25	15.60
	LeGO-LOAM [7]	0.061	12.37	9.19
Wells cave	**Proposed**	0.003	10.02	11.43
	HDL-NDT [17]	13.419	15.01	16.08
	HDL-Fast GICP [17]	1.07	12.48	11.91
	LeGO-LOAM [7]	Fail	8.11	8.18

TABLE III: LiDAR Inertial Odometry (LIO) comparison results in extreme environment datasets. (Red: Best, Blue: Second best)

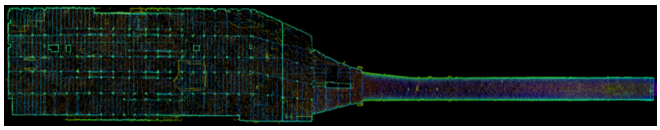
Dataset	Method	e2e Error [m]	Computational time [ms]	CPU [%]
Multi-floor	**Proposed**	0.142	16.92	11.16
	HDL-NDT [17]	Fail	20.23	12.41
	HDL-Fast GICP [17]	Fail	18.12	11.89
	LeGO-LOAM [7]	Fail	Fail	Fail
	LINS [12]	Fail	24.61	11.76
	LIO-SAM [8]	Fail	6.72	13.74
DARPA SubT Final Event	**Proposed**	0.0019	19.11	11.70
	HDL-NDT [17]	2.09	22.41	15.42
	HDL-Fast GICP [17]	6.93	19.57	12.99
	LeGO-LOAM [7]	Fail	Fail	Fail
	LINS [12]	Fail	Fail	Fail
	LIO-SAM [8]	Fail	8.48	12.78
		Mapping performance	**Computational time [ms]**	**CPU [%]**
Mega Cavern	**Proposed**	Reliable	9.25	10.14
	HDL-NDT [17]	Fail	30.24	15.67
	HDL-Fast GICP [17]	Distortion	23.25	14.15
	LeGO-LOAM [7]	Distortion	7.63	20.04
	LINS [12]	Fail	Fail	Fail
	LIO-SAM [8]	Reliable	10.18	21.41

method in Section III-E. All benchmark algorithms used for comparison were run on an Intel NUC computer with a 6 core i7-10710U CPU; Robot Operating System (ROS) in Ubuntu environment was implemented. Data was acquired in geometrically degraded and extreme environments such as multi-floor environments, narrow corridors, and huge caves as well as in feature dense environments. Details of datasets are shown in Table I.

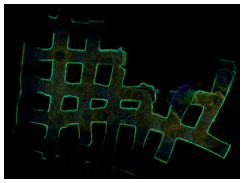
All datasets except for Mega Cavern were field test data of Team CoSTAR for the DARPA subterranean challenge final event; the Mega Cavern dataset was provided by courtesy of Team Explorer. It should be noted that the DARPA SubT Final Event dataset was acquired on the data collection day after the final competition. All datasets except for Mega Cavern were end-to-end (e2e) case with the same starting and ending point. We evaluated performance factors such as e2e error, average LiDAR odometry computational time per one LiDAR scan and average CPU load for comparison with benchmark algorithms. In the case of Mega Cavern dataset, because there was no ground truth and it was not an e2e case, we compared the map building performance for long term data.

B. Performance Comparison with Benchmark Algorithms

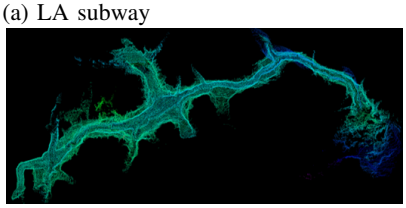
First, we evaluated the performance of the proposed method using only LiDAR in general environments with many edge and plane features, the comparison results are shown in Table II. HDL-graph-SLAM [17] used NDT [20] and Fast GICP [16] algorithms for scan matching as a point-based matching method. In the LA subway, Kentucky cave



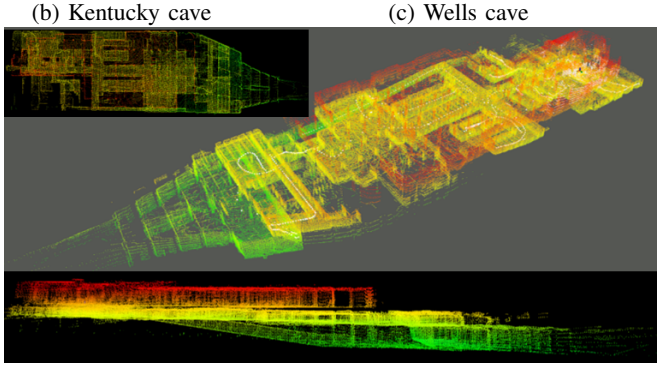
(a) LA subway



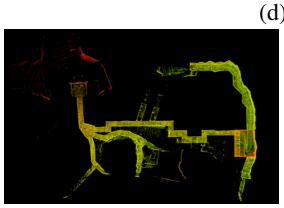
(b) Kentucky cave



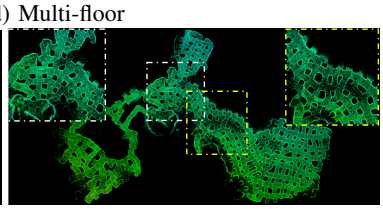
(c) Wells cave



(d) Multi-floor



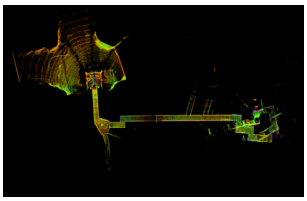
(e) DARPA SubT Final



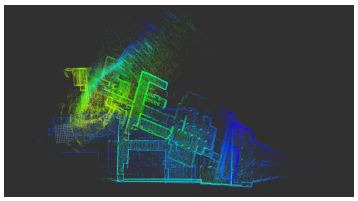
(f) Mega Cavern

Fig. 6: Maps generated with the proposed method in various datasets. Using both adaptive keyframe generation and sub-map matching, our method showed robust performance even in extreme underground environments

and Wells cave datasets, our method showed the highest accuracy among all benchmark algorithms with e2e errors of 0.041m, 0.044m and 0.003m, respectively. The LiDAR odometry computational time per one LiDAR scan of the proposed method were 16.33, 18.40 and 10.02ms, respectively, which were lower than those of HDL-graph-SLAM [17] and larger than those of LeGO-LOAM [7], which is a feature-based matching method, and CPU usage also showed a similar tendency. In the point-based matching method, the Fast GICP [16] method showed lower computational time than that of NDT [20] method. We also evaluated the performance of the proposed method in extreme environments; the comparison results are shown in Table III. For extreme environment datasets, only the proposed method succeeded in map generation. The proposed method showed high accuracy performance with errors of 0.142 and 0.0019m for the Multi-floor and DARPA SubT Final Event datasets, respectively. In terms of computational time and CPU load, similar to the results of general environment datasets, the proposed method performs better than those of the point-



(a) DARPA SubT Final map with LIO-SAM [8]



(b) Multi-floor map with HDL-Fast GICP [17]

Fig. 7: Distorted maps generated by conventional methods. Both feature-based method and the point-based method failed to provide reliable state estimation results in the extreme underground environment.

based matching method. However, depending on the features in extreme environments, our method sometimes showed lower values of computational time and CPU load performance than those of the feature-based matching method. In both general and extreme datasets, the feature-based matching method commonly fails to estimate the reliable state and to generate maps in geometrically degraded areas such as long tunnels, long corridors and narrow stairs. Finally, we evaluated the map generation performance of our method using the long term and long-distance Mega Cavern dataset with a total trajectory of 9118.11m. For the Mega Cavern dataset, only the proposed method and LIO-SAM [8] succeeded in generating a reliable map. Although LIO-SAM had map distortion in the early part, the map was corrected by the loop closure algorithm. On the other hand, our method, even though there is no loop closure algorithm, the map is reliable from beginning to end because the state at global scale is compensated for by *scan to sub-map* matching (as detailed in Section III-D). The maps generated by the proposed method are shown in Fig. 6 and the results of conventional methods that generate distorted maps in extreme underground environments are shown in Fig. 7.

V. Conclusions

In this paper we proposed an adaptive keyframe generation based hybrid LO/LIO algorithm that enables fast and accurate state estimation. In particular, unlike conventional LIO algorithms, adaptive keyframe generation according to the surrounding environment enables our method to balance between computational cost and accuracy. In our novel keyframe and map management module, sub-map generation is performed through the kNN method, which has low computational cost compared to the radius-based sub-map generation method. Compared with other state-of-the-art algorithms, the proposed method showed the most reliable state estimation performance. Finally, the proposed method was used in Team CoSTAR's autonomous flying drone Proxima in the DARPA subterranean challenge final event. For future work, we plan to add a feature extraction algorithm using a camera sensor to allow more accurate state estimation.

REFERENCES

[1] Tong Qin, Peiliang Li, and Shaojie Shen. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, 34(4):1004–1020, 2018.

[2] Antoni Rosinol, Marcus Abate, Yun Chang, and Luca Carlone. Kimera: an open-source library for real-time metric-semantic localization and mapping. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1689–1696, 2020.

[3] Michael Bloesch, Sammy Omari, Marco Hutter, and Roland Siegwart. Robust visual inertial odometry using a direct ekf-based approach. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 298–304, 2015.

[4] Zheyu Feng and Jianwen Li. Monocular visual-inertial state estimation with online temporal calibration. In *2018 Ubiquitous Positioning, Indoor Navigation and Location-Based Services (UPINLBS)*, pages 1–8, 2018.

[5] Ji Zhang and Sanjiv Singh. Loam: Lidar odometry and mapping in real-time. In *Proceedings of Robotics: Science and Systems (RSS '14)*, July 2014.

[6] H. Wang, C. Wang, C. Chen, and L. Xie. F-loam : Fast lidar odometry and mapping. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[7] Tixiao Shan and Brendan Englot. Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4758–4765, 2018.

[8] Tixiao Shan, Brendan Englot, Drew Meyers, Wei Wang, Carlo Ratti, and Daniela Rus. Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5135–5142, 2020.

[9] Christian Forster, Luca Carlone, Frank Dellaert, and Davide Scaramuzza. On-manifold preintegration for real-time visual–inertial odometry. *IEEE Transactions on Robotics*, 33(1):1–21, Feb 2017.

[10] Jorge J. Moré. The levenberg-marquardt algorithm: Implementation and theory. In G. A. Watson, editor, *Numerical Analysis*, pages 105–116, Berlin, Heidelberg, 1978. Springer Berlin Heidelberg.

[11] Haoyang Ye, Yuying Chen, and Ming Liu. Tightly coupled 3d lidar inertial odometry and mapping. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 3144–3150, 2019.

[12] Chao Qin, Haoyang Ye, Christian E. Pranata, Jun Han, Shuyang Zhang, and Ming Liu. Lins: A lidar-inertial state estimator for robust and efficient navigation. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8899–8906, 2020.

[13] Y. Chen and G. Medioni. Object modeling by registration of multiple range images. In *Proceedings. 1991 IEEE International Conference on Robotics and Automation*, pages 2724–2729 vol.3, 1991.

[14] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, pages 586–606. International Society for Optics and Photonics, 1992.

[15] Aleksandr Segal, Dirk Haehnel, and Sebastian Thrun. Generalized-icp. In *Robotics: science and systems*, volume 2, page 435. Seattle, WA, 2009.

[16] Kenji Koide, Masashi Yokozuka, Shuji Oishi, and Atsuhiko Banno. Voxelized gicp for fast and accurate 3d point cloud registration. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11054–11059. IEEE, 2021.

[17] Kenji Koide, Jun Miura, and Emanuele Menegatti. A portable three-dimensional lidar-based system for long-term and wide-area people behavior measurement. *International Journal of Advanced Robotic Systems*, 16, 02 2019.

[18] Masashi Yokozuka, Kenji Koide, Shuji Oishi, and Atsuhiko Banno. Litamin: Lidar-based tracking and mapping by stabilized icp for geometry approximation with normal distributions. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5143–5150. IEEE, 2020.

[19] Marius Muja and David G Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. *VISAPP (1)*, 2(331-340):2, 2009.

[20] Peter Biber and Wolfgang Straßer. The normal distributions transform: A new approach to laser scan matching. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*, volume 3, pages 2743–2748. IEEE, 2003.