



공학석사 학위논문

점별 라이다 불확실성 전파 및 연속 시간 궤적에 기반한 비동기 다중 라이다-관성 오도메트리

2023년 08월

서울대학교 대학원

기계공학부

정민우

점별 라이다 불확실성 전파 및 연속 시간 궤적에 기반한 비동기 다중

라이다-관성 오도메트리

지도 교수 김 아 영

이 논문을 공학석사 학위논문으로 제출함 2023년 04월

> 서울대학교 대학원 기계공학부 정 민 우

정민우의 공학석사 학위논문을 인준함 2023년 6월

위 위	원장.	이동준	(인)
부위	원장	김아영	(인)
위	원	한경원	(인)

초 록

최근 몇 년 동안, LiDAR는 주변의 거리를 정확하게 인식할 수 있다는 장점 덕분에 위치를 추정하거나 지도를 작성하는 과정에서 매우 중요한 센서로 사용되었으며, 이와 관련된 연구가 활발히 일어나고 있었다. 하지만 단일 LiDAR의 경우, 시야각(FOV) 문제나 물체의 가림 현상 등으로 인해 주변 인지에 한계가 존재하였고, 이는 기존의 목표를 수행하기엔 큰 단점으로 작용하였다. 이에, 높은 정확도와 확장된 FOV를 획득할 수 있는 다중 LiDAR 시스템은 대체 방안이 되었고, 기존의 문제를 해결할 수 있는 안정성 덕분에 해당 시스템이 인기를 끌게 되었다. 하지만, 다중 LiDAR를 통합하는 것은 시간적, 공간적 불일치로 인해 어려울 수 있다. 일반적인 방법은 엄격한 시간 동기화를 요구하면서 센서 간의 측정값인 점을 하나의 프레임으로 변환하거나, 센서 프레임 간의 변환을 근사화하여 모든 점을 하나의 프레임으로 통합하는 것이다. 기존의 방법들과는 다르게, 해당 논문에서는 IMU 이산 모델과 B-Spline Interpolation을 통한 연속 시간 궤적을 사용하여 센서 간의 시간적 변환을 상세히 설명하고, 이로 인해 발생된 불확실성을 각 점에 파생시킨다. 이런 불확실성은 상태 공분산을 취득 시간과 측정된 점의 거리와 결합하여 모델링함으로써, 다중 LiDAR를 사용할 때 엄격한 시간 동기화를 완화하고 FOV 차이를 극복할 수 있게 한다. 제안된 방법은 오픈 데이터셋과 직접 취득한 데이터셋에서 검증되었으며, 다양한 LiDAR 제조사와 스캔 패턴과 호환된다. 또한, 해당 논문은 LiDAR 관련 연구의 발전을 위해 Github에 공개하여, 다른 연구자들이 해당 연구에 접근할 수 있도록 소프트웨어를 오픈소스화 하였다.

주요어 : 다중 라이다, 슬램, 불확실성, 비동기 센서 융합 학 번 : 2021-24223

i

제	1 장 서	론	1
	제 1 절	연구의 배경	1
		제 1 항 시간적 불일치	2
		제 2 항 공간적 불일치	2
		제 3 항 불확실성 전파	3
	제 2 절	선행 연구 분석	5
		제 1 항 LiDAR-IMU SLAM	5
		제 2 항 다중 LiDAR-IMU SLAM	6
제	2 장 배	경 지식	7
.,	제 1 절	LiDAR 센서 원리	7
	제 2 절	강체 회전 및 프레임 변환	
	제 3 절	IMU Preintegration	9
	제 4 절	Iterated Error State Kalman Filter	11
제	3 장 디	·중 LiDAR-IMU 오도메트리	12
	제 1 절	시스템 소개	12
	제 2 절	연속 시간 궤적을 이용한 비동기 라이다 동기화	12
		제 1 항 IMU 이산 모델	12
		제 2 항 B-Spline 보간을 이용한 연속 시간 궤적 주정	13
	പി റ ചി	제 3 항 LiDAR 즉성값 보성 및 다중 라이다 동기화	14
	제 3 설	섬 멀 다이나 굴왁실성 선과	15
	세 4 설	Kalman Filter 기만 위지 좌석와	17
		세 1 양 나중 LIDAK 시스템 전저디 게 2 하 시거 미 츠저가 기바 이키 보고 미 0 카	17 10
		제 2 영 사진 및 특성없 기반 귀지 군포 및 오자	10 10
		제 5 중 위지 1 중을 위한 Localization 기 8시	19 19
	제 5 적	적 벽 북화식섯윽 고려하 맨핏	20
	AT U E		
제	4 장 실	험 결과	21
	제 1 절	실험 환경 소개	21
		제 1 항 데이터셋 소개	21
		제 2 항 평가 방법 소개	22
	제 2 절	실험 결과 분석	23
		제 1 항 Hilti 2021 SLAM Dataset	23
		제 2 항 UrbanNav Dataset	24
		제 3 항 City Dataset	26
	제 3 절	모뉼 별 기여도 분석	27
		제 1 항 모듈 별 정확도 분석	27
		제 2 항 지도 작성 기여 효과	29
		세 3 양 Localization 가중지 기여 효과	30
	세 4 설	나궁 다이나 개두 눈씩	31

32
21
21

표 목차

[표	1]	
- [표	2]	
[표	3]	
[표	4]	
[표	5]	
[표	6]	

그림 목차

[그림	1]	1
[그림	2]	4
[그림	3]	.14
[그림	4]	.18
[그림	5]	.23
[그림	6]	.25
[그림	7]	.25
[그림	8]	.27
[그림	9]	. 29
[그림	10]	. 30
[그림	11]	. 31
[그림	12]	. 32

제1장서 론



그림 1: (a) The B-Spline interpolation allows for a precise consolidation of points, which expands the overall field of view (FOV) in the composite map (b) Two distinct types of uncertainties are modeled (c) An illustration of the FOV variations in datasets incorporating multiple LiDARs.

제 1 절 연구의 배경

지난 수십 년 동안, LiDAR를 활용한 로봇 네비게이션 및 Simultaneous Localization and Mapping (SLAM)은 위치 확인과 지도 구성 분야에서 상당한 발전을 이루어 왔다. 대부분의 기존 방법들은 단일 LiDAR 시스템을 이용한 어플리케이션을 제시했지만, 제한된 시야(FOV)와 물체의 가림 현상 등의 문제는 하드웨어에 기반한 문제이기 때문에 하나의 LiDAR만으로는 해결할 수 있지 않았고 다중 LiDAR의 필요성이 대두되었다. 다중 LiDAR의 경우, 취득된 모든 점을 하나로 합쳐 기존 단일 LiDAR의 프레임워크에 사용할 수도 있지만, 근본적으로 다중 LiDAR에서 취득된 점을 하나로 통합할 때, 시간과 공간 차이라는 두 가지 큰 도전 과제가 존재하며, 이들은 각 LiDAR에서 취득된 점군의 통합을 방해하게 된다.

제 1 항 시간적 불일치

첫째로, 시간적 동기화 문제다. 센서들 간에 엄격한 시간 동기화를 구현하는 것이 직관적이고 간단해 보이지만, 실제로는 복잡하다. 센서들은 외부 하드웨어를 통해 Pulse per Second(PPS) 신호를 전송하여 동기화할 수 있으나, 이는 추가적인 하드웨어가 필요하다는 뜻이며, LiDAR의 개수가 많아진다면 또다른 문제로 야기될 수 있다. PPS를 사용하지 않더라도 대안으로 Precision Time Protocol(PTP)를 사용할 수 있지만, PTP의 경우 모든 제조사가 이를 지원하지는 않아 동일한 제조사에서 제작한 센서 조합이 필요하다. 또한, PTP는 각 센서의 시간을 하나의 기준 프레임으로 통일해주는 것이지, 모든 센서의 측정 시간을 하나의 시점으로 통일해주지 않는다는 점에서 차이가 존재한다. 이처럼 복잡한 동기화 설정 때문에, 오픈 데이터셋들 [1-4]에서도 다중 LiDAR의 엄격한 시간 동기화는 존재하지 않으며, 따라서 시간 차이를 처리하기 위한 소프트웨어 해결책이 필요하다. 이런 시간적 동기화를 위해, IMU를 활용하여 시간 차이를 보상한 연구 [5, 6]도 존재하며, IMU를 활용하는 아이디어는 실용적이었지만 IMU 모델이 이산 전파에 비롯되며, 이로 인해 발생한 오류는 여전히 존재했다. 또한, 몇몇 연구자들 [7]은 연속 시간 궤적 추정을 위해 최적화를 사용하는 것을 제안했으나 이 접근법은 각 시간의 어느 지점에서나 전체 궤적을 추정할 수 있지만, 강한 계산 소모 값 때문에 실시간 가능성이 제한되었다.

제 2 항 공간적 불일치

두 번째로, 공간적 차이에 기반한 문제이다. 앞서 언급된 시간적 차이를 해결하는 또 다른 방법은 스캔 매칭을 적용하고 시간적 차이에 해당되는 움직임만큼 계산하여 프레임 변환 시 보완하는 것이다. 하지만, 다른 스캔 패턴과 FOV로 인해 생기는 공간적 차이는 스캔 매칭이 시간 차이를 보완하는 것을 방해한다. 특히, 반복되지 않는 스캔 패턴으로 인한 점군의 낮은 밀도와 FOV 차이 [1], 그리고 비스듬하게 설치된 회전 형태의 LiDAR [2]는 각 센서들 사이에 겹치는 영역을 매우 작게

제 3 항 불확실성 전파

만약 앞선 1항과 2항의 문제가 해결되었다고 하더라도 시간과 공간의 차이로 인해 센서들 사이의 점군을 각 프레임으로 투영하는 동안 불가피하게 불확실성을 축적할 수밖에 없다. 따라서, 이 문제에 대해서는, 적절한 불확실성 모델링을 통해 다중 LiDAR들 사이의 전달된 모호성을 포착할 수 있어야 한다. 최근 몇 년 동안, LiDAR에서의 불확실성은 주로 딥러닝 기반의 방법들에 초점을 맞추어 광범위하게 연구 [8-11]되어 왔다. 그러나, 해당 논문과 같이 일부 모델 기반 접근법들도 제시되었는데, 예를 들어 Wang et al. [6]은 IMU와 LiDAR 오도메트리 사이의 차이를 바탕으로 LiDAR 잔차의 가중치를 계산하였지만, 모든 점을 동등한 가중치로 다루었다. 해당 논문에서는 각 점의 범위와 취득 시간을 고려하여 점별 불확실성을 부여한다는 점에서 차이가 있으며, 시간적 및 공간적 차이에 의해 유발된 모호성을 모두 다루는데 특히 중요하다. M-LOAM [12]은 외부 공분산과 상태 공분산을 사용하여 각 점에서의 불확실성을 전파했다. 이 접근법은 상태 공분산과 점별 불확실성 전파를 활용한다는 점에서 해당 논문과 유사하나, M-LOAM과 달리, 해당 논문은 외부 공분산 업데이트를 위한 LiDAR 간의 공통된 스캔 영역이 필요하지 않으므로, 점 수준의 불확실성을 더 일반적으로 처리하는 데 유리하다. 추가적으로, 불균형한 지형을 다루기 위해 불확실성을 사용하여 일반 방향의 분산을 고려한 연구 [13]도 존재하나, 취득 시간과 범위 모호성을 통합하지 않았다.

기존의 방법들과 달리 해당 논문은 비동기 다중 LiDAR-관성 오도메트리를 제안한다 [그림 2]. 다중 LiDAR에서 언급된 시간적, 공간적 도전 과제를 다루기 위해, 우리는 범위와 각 시간의 상태 공분산을 고려하여 점 별 불확실성을 모델링한다. 또한, 주변에 인식된 3D 환경을 활용하여 최적화 중에 가중치 항을 결정하는 Localization 가중치를 계산한다. 다중 LiDAR에서 처리되는 대량의 점들을 효과적으로 처리하기 위해, ikd-Tree와 Iterative Error State Kalman Filter (IESKF) [14]를 활용하여 계산 효율성을 달성했다. 해당 논문의 핵심 사항은 아래와 같다.

• FOV 차이로 인해 스캔 매칭으로 해결하지 못한 시간적 보상은 연속 시간 B-Spline Interpolation을 활용하여 구함으로써, 심각한 FOV 차이에도 불구하고 비동기 LiDAR 스캔 정렬의 일관성을 유지했다.

- 제안된 점 수준의 불확실성은 범위와 점 취득 시간에 의해 증가된 불확실성을 포착한다. 이 점 별 불확실성은 더 멀리 또는 더 늦게 취득된 점에 대해 더 큰 불확실성을 부여함으로써, 불확실성을 더 일반적으로 다루도록 하였다.
- 제안된 Localization 가중치는 최적화 중에 사전 분포와 측정값 분포 사이의 비율을 조정한다. 이를 통해 터널이나 좁은 복도와 같은 어려운 환경에서 각 분포의 비율을 자동으로 조정할 수 있다.
- 제안된 방법은 오픈 데이터셋과 직접 취득한 데이터셋에서 검증되었으며, 다양한 제조사에서 다른 스캔 패턴을 가진 어떤 LiDAR 조합과도 호환된다. 또한 각 모듈의 의의성과 LiDAR 개수에 따른 시간 및 정확도를 추가로 분석하였다.



그림 2: The proposed method is made up of three parts: preprocessing, state estimation, and mapping. In the preprocessing, points from each LiDAR are adjusted to remove distortion and combined using B-spline interpolation. During the state estimation phase, we propagate uncertainty on a point-wise basis and use an IESKF until a state of convergence is reached. Lastly, the optimal state is fed into the IMU model for precise interpolation in future scans. After evaluating uncertainty, the points are then projected into an ikd-Tree for organizing points in a k-dimensional space.

제 2 절 선행연구 분석

제 1 항 LiDAR-IMU SLAM

해당 항에서는 LiDAR-IMU SLAM의 선행 연구에 대해 다룬다. LiDAR-IMU SLAM은 IMU의 사용에 따라 loosely-Coupled와 tightly-Coupled로 구분할 수 있으며, 전자의 경우 IMU는 초기 상태 추정에만 사용되고 최적화 과정 시에는 사용되지 않는다는 특징이 있다. 이는 빠른 최적화가 가능하다는 장점이 있지만, 제약 조건이 사라지는 만큼 정확도가 떨어질 수 있다는 단점이 있으며, 후자의 경우 이와 반대되는 특성을 가지고 있다.

LOAM [16]은 특징점을 찾기 위해 면과 가장자리를 활용하며, 특징점을 검출하기 전에 IMU를 이용해 포인트 클라우드를 보정한다. 그러나 이 방법은 특징점을 검출하는 데 상당한 시간이 소요된다는 문제점이 있다. 이러한 문제는 LeGO-LOAM [17]에서 개선되었으며, 이 방법은 LOAM과 대체적으로 같은 방식을 사용하지만, Levenberg-Marquardt 알고리즘을 사용해 계산 시간을 줄이고, Loop Closing을 추가해 SLAM의 안정성을 높였다. 두 방법 모두 IMU를 이용해 왜곡을 없애고 높은 정확도로 움직임을 추정할 수 있지만, loosely-coupled로 인해 큰 정확도는 얻지 못했다.

이와 달리, IN2LAMA [18], LIOM [19], LIRO [20], LIO-SAM [21]등의 방법에서는 IMU 및 다른 센서를 더욱 적극적으로 활용해 정확도를 크게 향상시켰다. 이들 방법은 IMU Preintegration을 사용해 두 프레임 사이의 IMU 값들을 하나의 상대적 위치 변화로 계산하여, 계산의 반복과 불필요한 변수들을 줄여 계산 복잡성을 해결하는 데 도움을 주었다. 특히 LIO-SAM은 그래프 최적화를 이용하여 더더욱 안정성을 높였다.

해당 시점부터는 점차 다양한 방법으로 LiDAR-IMU SLAM의 정확도 및 처리 속도를 올리게 되는데, 연속 시간 궤적을 추정하고 최적화 하는 CLINS [7]나 Wildcat[22]이 등장하기도 하였으며, 칼만 필터를 기반으로 속도와 정확도를 모두 달성한 Fast-LIO [14, 23], Faster-LIO [24], 각 스캔을 여러 프레임으로 나누어 빈번하게 시스템의 상태를 추정하면서 정확도를 높이는 FR-LIO [25]나 Point-LIO [26]등이 등장하였다. 그 외에 LiDAR 점군에서 동적 물체를 제거 [27]하거나 LiDAR 점의 반사 강도를 함께 이용해 최적화하는 연구 [28]도 이루어지는 등 매우 많은 시도가 있었지만, 해당 방법들은 모두 LiDAR 하나에 의존하고 있다는 단점이 존재한다. 단일 LiDAR가 하드웨어 문제에 처한다면 바로 위치 추정에 실패할 것이며, 일반적인 상황에서도 하드웨어의 종류에 따라 문제가 발생할 수 있다. 따라서 최근에는 각각의 단일 LiDAR의 단점을 상호 보완할 수 있는 다중 LiDAR-IMU SLAM이 연구되고 있다.

제 2 항 다중 LiDAR-IMU SLAM

앞서 언급되었듯, 다중 LiDAR는 넓은 FOV를 관측할 수 있어 많은 측정값을 가질 수 있고, 각각의 스캔 패턴에 따라 밀도가 높은 점군을 획득해 주변 환경의 세세한 특징도 파악할 수 있다는 장점이 있다. M-LOAM [12]은 다중 회전 LiDAR를 동기화하여 주변의 점군을 획득하고 위치를 추정하였다. 이와 동시에 실시간 캘리브레이션을 지원하여 각 LiDAR 사이의 extrinsic parameter를 구할 수 있었지만, 엄격한 시간적 동기화는 다른 제품의 LiDAR에서는 힘들 수 있으며, M-LOAM 역시 모든 실험이 최대 4개의 동일한 회전 LiDAR에서 이루어졌다. LOCUS [29]에서도 시간적 동기화를 가정하였으며, LOCUS 2.0 [4]에서는 지연된 LiDAR를 위치 추정에 사용하지 않았다. 이는 정확도를 높이는데 도움이 될 수는 있으나, 짧은 시간에 빠른 속도로 움직인다면 문제가 되는 것은 동일하며, 유용할 수 있는 정보가 손실되었다고 볼 수도 있다. MiLIOM [5], SLICT [30] 및 D-LIOM [31] 경우 IMU를 기반으로 시간적 동기화를 진행하였지만, 해당 과정에서 IMU 이산 모델과 선형 근사를 사용하여 정확도가 떨어지는 편이다. 이 외에도 다중 LiDAR에 대한 연구들 [6, 32]이 추가로 존재하지만 모두 IMU 이산 모델로 인한 오류가 남아 있으며, 모두 1장에서 언급된 시간에 따른 불확실성에 대한 고려가 존재하지 않는다. 제안된 방법은 비동기 스캔을 포함한 모든 가능한 정보를 활용하고, 선형 근사를 피한다. 실시간 프로세스를 위해 최적화 단계에서 control points를 포함시키지 않는 대신에, IMU 이산 모델을 기반으로 B-Spline Interpolation을 수행해 각 LiDAR 측정치에 대한 궤적을 추정하였다. 이를 통해 다중 LiDAR SLAM에 대한 효율적인 해결책을 제공한다

제 2 장 배경 지식

해당 장에서는, LiDAR 오도메트리를 위한 센서의 원리, 수학적 계산 등의 기초적인 배경 지식에 대해 설명한다.

제 1 절 LiDAR 센서 원리

LiDAR는 Light Detection and Ranging의 줄임말로, 빛을 이용해 거 리를 측정하는 센서이다. LiDAR는 레이저를 이용해 물체까지의 거리를 측정하고, 그 측정 데이터를 이용해 3차원 공간을 구성할 수 있다. 이 기술은 다양한 분야에서 활용되고 있으며, 특히 자율주행 차량, 드론, 로 보틱스 등에서 중요한 역할을 하고 있다. LiDAR는 크게 두 가지 형태로 나뉘는데, 하나는 Spinning LiDAR, 다른 하나는 Solid State LiDAR이다.

Spinning LiDAR, 회전 LiDAR는 물리적으로 회전하면서 레이저를 발사하고 그 반사를 검출하는 방식으로 작동한다. 이 센서는 높은 해상 도와 넓은 FOV를 제공하지만, 물리적으로 회전해야 하기 때문에 크기가 상대적으로 크고 내구성에 문제가 있을 수 있다. 또한, 회전 속도에 따 라 감지하는 점의 밀도가 달라진다.

Solid State LiDAR는 물리적인 이동 부분 없이 전자적, 광학적 방법 으로 레이저를 스캔하는 기술이다. 이 센서는 물리적인 부품이 움직이지 않기 때문에 내구성이 뛰어나고, 크기가 작아 자동차나 드론 등에 적용 하기 용이하다. 하지만 상대적으로 회전 LiDAR에 비해 시야각이나 해상 도가 떨어지는 경우가 많다.

각 LiDAR 센서의 측정 방식은 레이저를 특정 방향으로 발사한 후 거울 또는 프리즘을 거쳐, 물체에 반사되어 돌아오는 레이저를 감지하여 레이저가 이동하는 시간을 측정하고, 이를 통해 물체까지의 거리를 계산한다. 이 방식을 이용하면 물체의 모양, 위치 뿐만 아니라 최근에는 이동 속도 등을 파악할 수도 있다. 또한 레이저가 거치는 광학 편광 요소에 따라 특정 스캔 패턴을 생성시킬 수 있으며, 분산 프리즘이나 회절 격자는 같은 축을 회전하여 패턴이 반복되지 않게 만들 수도 있다. 이를 통해 정지해 있을 경우 검출하는 영역의 밀도가 높아지며, 센서의 FOV 내에서 세세한 특징들을 잡을 수 있다. 일반적으로 LiDAR 센서는 수십 또는 수백 개의 레이저를 동시에 발사하고 감지할 수 있어, 높은 해상도의 3차원 데이터를 얻을 수 있다.

제 2 절 강체 회전 및 프레임 변환

각 LiDAR에서 검출된 점들은 3D 공간상에 위치하게 되는데, 이 때 (x, y, z)를 표현하는 기준은 local 좌표계, 즉 LiDAR 좌표계가 된다. 하지만 이런 점은 어떤 좌표계로 표현하는지에 따라 위치가 바뀔 수 있는데, 이러한 위치 관계를 표현하는 것이 변환 행렬이다. 이번 절에서는 이러한 변환 행렬에 대해서 다루게 된다.

 SO(3) = {R ∈ ℝ^{3×3} | RR^T = I, det(R) = I }
 (2.1)

 (2.1)은 Special orthogonal group이라 불리는 SO3에 대한

 정의이다. 3×3 의 크기로 구성이 된 정사각행렬이며, 해당 행렬의

 역행렬은 전치행렬과 동일하다는 특성을 가지고 있다. SO3 그룹은

 곱셈에 대해 닫혀 있기 때문에, SO3 행렬 간의 곱은 SO3 행렬, 즉 회전

 행렬에 해당하는 결과를 도출한다.

 $SE(3) = \left\{ \mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{bmatrix} \in \mathbb{R}^{4 \times 4} \mid \mathbf{R} \in SO(3), \mathbf{t} \in \mathbb{R}^3 \right\}$ (2.2)

(2.2)의 경우 Special Euclidean 3 (SE(3)) 그룹이라 불리며, Lie group에 속한다. SE3는 SO3인 회전 행렬과 3차원 t 벡터를 포함하는 4×4 의 정사각행렬이며, 3D 공간에서의 강체 변환을 할 수 있는 행렬이다. SO3 행렬과 마찬가지로 곱셈에 대해 닫혀 있으며, 따라서 SE3 그룹 간의 곱은 여전히 SE3 그룹의 결과를 도출한다. 이와 같은 행렬은 다음과 같이 사용된다.

$T_A^C = T_B^C T_A^B \tag{2.3}$

 T_A^B 를 A 좌표계에서 B 좌표계로 변환하는 행렬이라고 정의할 때, T_B^C 는 B 좌표계에서 C 좌표계로 변환하는 행렬을 의미한다. 이 둘을 이용하여 (2.3)과 같이 계산을 한다면 T_A^C , 즉 A에서 C 좌표계로 변환하는 행렬이 계산된다. 이를 A LiDAR에서 검출된 점, P_A 에 적용시켜 B 좌표계에서의 점 P의 위치를 표현한다면 P_B 는 다음과 같이 계산된다.

$$P_B = T_A^B P_A \tag{2.4}$$

이러한 SE3 그룹은 항상 역행렬을 가지며, 이러한 역행렬은 좌표계 변환의 반대 과정과 동일하다. 즉 $T_A^B = (T_B^A)^{-1}$ 와 동일하며, 각 행렬의 SO3와 3차원 t 벡터 사이의 관계는 아래와 같다.

$$T_A^B = \left(T_B^A\right)^{-1} = \begin{pmatrix} (\mathbf{R}_B^A)^T & -(\mathbf{R}_B^A)^T \mathbf{t}_B^A \\ 0 & 1 \end{pmatrix}$$
(2.5)

제 3 절 IMU Preintegration

IMU의 경우, 카메라나 LiDAR에 비해 짧은 주기성을 가진 센서로 각 시점에서의 3차원 각속도 값과 가속도 값을 측정하는 센서이다. IMU 센서의 중요한 역할 중 하나는 상대적으로 긴 주기의 센서 측정값 사이를 연결해줄 수 있다는 점이다. 총 6 DOF의 측정 값을 바탕으로, 해당 값을 적분하여, 각 시점에서의 위치나 속도 정보 등을 구할 수 있지만 이렇게 모든 지점에서의 상태를 추정하고, 후에 카메라 또는 LiDAR의 측정값과 함께 최적화된다면 매개변수의 수가 크게 증가하면서 위치 추정의 핵심 중 하나인 실시간성을 잃을 수 있게 된다. 따라서 해당 절에서 소개하는 IMU Preintegration의 목표는 모든 IMU 측정값을 하나의 측정값으로 합치는 과정을 보여준다. 즉, 상대적으로 늦은 측정 주기를 지닌 LiDAR와 카메라 사이의 상대적 움직임을 계산하기 위해 IMU 측정값을 하나로 합쳐야 한다.

IMU의 경우 소자의 특성이나 불순물 등으로 측정값에 bias 또는 white noise 등이 존재하는데, 이를 수식으로 작성하면 아래와 같다.

 ${}_{I}\widetilde{\boldsymbol{\omega}}(t) = {}_{I}\boldsymbol{\omega}(t) + \mathbf{b}^{g}(t) + \boldsymbol{\varphi}^{g}(t)$ (2.6)

 $_{I}\tilde{\mathbf{a}}(t) = \mathbf{R}_{WI}^{T}(t)(_{W}\mathbf{a}(t) - _{W}\mathbf{g}) + \mathbf{b}^{a}(t) + \boldsymbol{\varphi}^{a}(t)$ (2.7) 이 때, 위 첨자 g, a는 자이로스코프와 가속도를 뜻하며, 아래 첨자 \mathbf{I}, \mathbf{W} 는 각각 IMU 좌표계와 전역 좌표계를 의미한다. 마지막으로 (2.6)에서의 \mathbf{g} 는 중력 가속도를 의미한다. 좌측의 항은 측정값이며, 우측의 항은 참값과 bias, white noise를 분리시킨 것을 보여준다. 위치, 속도, 가속도의 관계성과 각도와 각속도의 관계성을 활용하면, 해당 식을 바탕으로 연속적인 시간에 관한 식이 도출된다.

$$\mathbf{R}_{WI}(t + \Delta t) = \mathbf{R}_{WI}(t) \operatorname{Exp}\left(\int_{t}^{t + \Delta t} {}_{I}\boldsymbol{\omega}(\tau) \, d\tau\right)$$
(2.8)

$${}_{W}\mathbf{v}(t+\Delta t) = {}_{W}\mathbf{v}(t) + \int_{t}^{t+\Delta t} {}_{W}\mathbf{a}(\tau) d\tau$$
(2.9)

 ${}_{W}\mathbf{p}(t+\Delta t) = {}_{W}\mathbf{p}(t) + \int_{t}^{t+\Delta t} {}_{W}\mathbf{v}(\tau) d\tau + \iint_{t}^{t+\Delta t} {}_{W}\mathbf{a}(\tau) d\tau \quad (2.10)$

는 (2.5)와 (2.6)의 식을 바탕으로 적분한 식이며, 실제로 IMU의 경우 연속적이지 않고 이산적이기 때문에, (2.7-2.9)에 해당하는 식을 이산 모델로 정리하면 아래와 같다.

$$\mathbf{R}_{WI}(t + \Delta t) = \mathbf{R}_{WI}(t) \operatorname{Exp}(I\boldsymbol{\omega}(\tau)\Delta t)$$
(2.11)

$${}_{W}\mathbf{v}(t+\Delta t) = {}_{W}\mathbf{v}(t) + {}_{W}\mathbf{a}(\tau)\Delta t \qquad (2.12)$$

$${}_{W}\mathbf{p}(t+\Delta t) = {}_{W}\mathbf{p}(t) + {}_{W}\mathbf{v}(t)\Delta t + \frac{1}{2}{}_{W}\mathbf{a}(\tau)\Delta t^{2}$$
(2.13)

(2.8-2.10)에서 구한 식에 noise와 white noise를 포함하고, 앞서 언급했던 것과 같이 특정 두 시점, *i*과 *j* 사이에 관계성을 구하기 위해 해당 식을 축적하면 아래와 같다.

$$\mathbf{R}_{j} = \mathbf{R}_{i} \prod_{k=i}^{j-1} \exp\left(\left(\widetilde{\boldsymbol{\omega}}_{k} - \mathbf{b}_{k}^{g} - \boldsymbol{\varphi}_{k}^{g}\right) \Delta t\right)$$
(2.14)

$$\mathbf{v}_{j} = \mathbf{v}_{i} + \mathbf{g}\Delta t_{ij} + \sum_{k=i}^{j-1} \mathbf{R}_{k} (\tilde{\mathbf{a}}_{k} - \mathbf{b}_{k}^{a} - \boldsymbol{\varphi}_{k}^{a}) \Delta t \qquad (2.15)$$

$$\mathbf{p}_{j} = \mathbf{p}_{i} + \sum_{k=i}^{j-1} \left[\mathbf{v}_{k} \Delta t + \frac{1}{2} \mathbf{g} \Delta t^{2} + \frac{1}{2} \mathbf{R}_{k} (\tilde{\mathbf{a}}_{k} - \mathbf{b}_{k}^{a} - \boldsymbol{\varphi}_{k}^{a}) \Delta t^{2} \right] \quad (2.16)$$

이를 바탕으로, 상대적 모션 증분을 구할 수 있는데, 여기서 구하는 모션 증분은 실제 물리적인 증분과 동일하지 않으며, 그 이유는 전개의 편리성을 위하여 중력과 관련된 항을 이항하여 전개하였기 때문이다. 결과적으로 각 증분은 아래와 같이 계산된다.

$$\Delta \mathbf{R}_{ij} = \mathbf{R}_i^T \mathbf{R}_j = \prod_{k=i}^{j-1} \operatorname{Exp}\left(\left(\widetilde{\boldsymbol{\omega}}_k - \mathbf{b}_k^g - \boldsymbol{\varphi}_k^g\right) \Delta t\right)$$
(2.17)

$$\Delta \mathbf{v}_{ij} = \mathbf{R}_i^T (\mathbf{v}_j - \mathbf{v}_i + \mathbf{g} \Delta t_{ij}) = \sum_{k=i}^{j-1} \mathbf{R}_{ik} (\tilde{\mathbf{a}}_k - \mathbf{b}_k^a - \boldsymbol{\phi}_k^a) \Delta t \quad (2.18)$$

$$\Delta \mathbf{p}_{ij} = \mathbf{R}_i^T \left(\mathbf{p}_j - \mathbf{p}_i - \mathbf{v}_i \Delta t_{ij} - \frac{1}{2} \sum_{k=i}^{j-1} \mathbf{g} \Delta t^2 \right)$$

=
$$\sum_{k=i}^{j-1} \left[\mathbf{v}_{ik} \Delta t + \frac{1}{2} \mathbf{R}_{ik} (\tilde{\mathbf{a}}_k - \mathbf{b}_k^a - \boldsymbol{\phi}_k^a) \Delta t^2 \right]$$
 (2.19)

최종적으로 전개를 통해 보이고 싶은 것은 계산된 상대적 모션 증분이 추정해야 하는 모션 증분과 이에 대한 랜덤 백터의 합으로 표현이 가능하다는 것이다. 이에 대한 전개는 [15]를 통해 확인할 수 있으며, 두 프레임 사이에 상대적 증분은 다음과 같은 표현으로 정리될 수 있다.

$$\Delta \mathbf{R}_{ij} = \mathbf{R}_i^T \mathbf{R}_j \operatorname{Exp}(\delta \Phi_{ij})$$
(2.20)

$$\Delta \mathbf{v}_{ij} = \mathbf{R}_i^T (\mathbf{v}_j - \mathbf{v}_i + \mathbf{g} \Delta t_{ij}) + \delta \mathbf{v}_{ij}$$
(2.21)

$$\Delta \mathbf{p}_{ij} = \mathbf{R}_i^T \left(\mathbf{p}_j - \mathbf{p}_i - \mathbf{v}_i \Delta t_{ij} - \frac{1}{2} \mathbf{g} \Delta t_{ij}^2 \right) + \delta \mathbf{p}_{ij}$$
(2.22)

제 4 절 Iterated Error State Kalman Filter

이번 절에서 다룰 Iterated Error State Kalman Filter [14]는 기존 의 Kalman Filter와 다르게 각 상태의 에러를 추정하는 칼만 필터 방식 이다. 기본적으로 상태 추정 시 추정 값 (nominal state)와 에러 값 (error value)를 더해 참값 (true value)를 구할 수 있다. 이 때 Error State Kalman Filter는 더해지는 에러 값을 추정하고, 처리하는 방식을 의미한다. 에러 상태는 주로 값이 작기 때문에 칼만 필터에 비해 2차항 과 같은 높은 차수의 값을 쉽게 무시할 수 있다는 장점이 있으며, 몇몇 자코비안 역시 상수항으로 두어도 문제가 없다는 특징이 있다. 즉 선형 성을 가지고 있다고 볼 수 있으며, 이와 같은 특징으로 인해 상태 추정 시 비선형성을 가진 상태를 추정하는 기존 칼만 필터에 비해 좀 더 정확 하게 작동할 수 있다.

Iterated Error State Kalman Filter는 다음과 같은 절차로 동작하는데, 먼저 해당 상태의 추정 값을 IMU Preintegration 또는 주어진 오도메트리로 예측하고, 이에 해당하는 공분산을 구한다. 그 후, 예측된 상태를 바탕으로 측정 값에 대한 에러 상태와 칼만 게인을 계산한다. 그 후 에러 상태와 추정된 상태를 통해 시스템의 상태를 보정하고, 공분산 값을 업데이트 한다. 이 때 업데이트는 계속 반복하여 이전 업데이트와의 차이가 특정 임계 값 이하가 되면 멈춘다.

이와 같은 과정을 진행할 때, 에러 상태는 원점에서만 동작한다. 즉, 에러 상태는 평균이 0인 가우시안 분포를 따른다. 이러한 성질 덕분에 최적화 시 파라미터에 종속적인 현상이 발생하지 않고, 앞서 언급된 선형성과 더불어 항상 선형화를 진행할 수 있다. 회전에 대해서도 에러 상태가 SO3에 대한 파라미터를 가지기 때문에, 최소한의 파라미터를 가지며 위에서 언급된 것과 같이 짐벌락 현상이 발생하지 않는다.

제 3 장 다중 LiDAR-IMU 오도메트리

해당 장에서는 논문의 주 내용이 되는 다중 LiDAR-IMU 오도메트리를 위한 B-Spline Interpolation과 불확실성을 각 LiDAR 점 별로 전파하는 과정, Localization 가중치, 그리고 사전 분포 모델과 다중 LiDAR 측정 분포 모델을 기반으로 한 Iterative Error State Kalman Filter를 통해 어떻게 시스템의 상태를 추정하는지 소개한다.

제 1 절 시스템 소개

문자 오른쪽 아래 첨자 ()_A 의 경우 frame A 에 대해 표시하고 있음 을 뜻한다. 또한 ()_{AB}는 frame A 에서 바라본 frame B 를 의미한다. 참값 및 전파 값, 에러 값, 최적 값은 각각 (),(),(),() 로 표기되었다. 명료성을 위해, N 개의 LiDAR 중 가장 마지막 시간대에 점을 취득한 LiDAR 를 Primary LiDAR (P), 그리고 그 외의 모든 LiDAR 를 Secondary LiDAR (S)로 표기하였다. 제안한 시스템은 state **x**와 input **u**, and noise **w**로 구성된다.

$$\mathcal{M} \triangleq SO(3) \times \mathbb{R}^{15} \times \Pi_{i=1}^{N} \left(SO(3) \times \mathbb{R}^{3} \right)$$
$$\mathbf{x} \triangleq \left[\mathbf{R}_{GI}^{T} \quad \mathbf{t}_{GI}^{T} \quad \mathbf{v}_{GI}^{T} \quad \mathbf{b}_{\omega}^{T} \quad \mathbf{b}_{a}^{T} \quad \mathbf{g}_{G}^{T} \quad \mathbf{R}_{IL_{i}}^{T} \quad \mathbf{t}_{IL_{i}}^{T} \right]^{T} \in \mathcal{M}$$
$$\mathbf{u} \triangleq \left[\boldsymbol{\omega}_{m}^{T} \quad \boldsymbol{a}_{m}^{T} \right]^{T}, \mathbf{w} \triangleq \left[\boldsymbol{n}_{\omega}^{T} \quad \boldsymbol{n}_{a}^{T} \quad \boldsymbol{n}_{b\omega}^{T} \quad \boldsymbol{n}_{ba}^{T} \right]^{T}.$$
(3.1)

Global frame (G)에서 본 IMU frame (I)의 변환 행렬은 $\mathbf{T}_{GI} = (\mathbf{R}_{GI}, \mathbf{t}_{GI})$ 이며, 추가로 **v**,**g**, **b**는 각각 속도와 중력, 그리고 편향을 의미한 다. \mathbf{T}_{IL} 은 IMU와 LiDAR 사이의 extrinsic calibration 이다. 또한 ω_m 및 a_m 는 IMU 로부터 취득된 각속도와 가속도이며, w는 해당 변수들에 대한 화이트 노이즈를 포함한다.

제 2 절 연속 시간 궤적을 이용한 비동기 라이다 동기화

제 1 항 IMU 이산 모델

연속 시간 IMU 모델은 [14]에 소개된 田 연산자로 다음과 같이 이산 모델로 변환할 수 있다

$$\mathbf{x}_{i+1} = \mathbf{x}_{i} \boxplus (\Delta \mathbf{t} \mathbf{f}(\mathbf{x}_{i}, \mathbf{u}_{i}, \mathbf{w}_{i})), \qquad (3.2)$$

$$\mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{w}) = \begin{bmatrix} \boldsymbol{\omega}_{m} - \boldsymbol{b}_{\omega} - \boldsymbol{n}_{\omega} \\ \mathbf{v}_{GI} + \frac{1}{2} (\mathbf{R}_{GI} (a_{m} - \boldsymbol{b}_{a} - \boldsymbol{n}_{a}) + \mathbf{g}_{G}) \Delta \mathbf{t} \\ \mathbf{R}_{GI} (a_{m} - \boldsymbol{b}_{a} - \boldsymbol{n}_{a}) + \mathbf{g}_{G} \\ \boldsymbol{n}_{b\omega} \\ \boldsymbol{n}_{ba} \\ \boldsymbol{0}_{3 \times 1} \\ \boldsymbol{0}_{3 \times 2N} \end{bmatrix}$$

여기서 f는 시스템의 이전 또는 다음 state 를 연결시켜주며, 시간 Δt를 바탕으로 파라미터화가 가능하다. LiDAR 의 (*i* – 1)th과 (*i*)th 사이의 스캔에서 (*i* – 1)th에서의 state 가 최적화된 값이라면, IMU 를 이용해 추정한 경로는 다음과 같다.

$$\widehat{\mathbf{x}}_{k+1} = \widehat{\mathbf{x}}_k \boxplus (\Delta \mathbf{t} \mathbf{f}(\widehat{\mathbf{x}}_k, \mathbf{u}_k, \mathbf{0})); \widehat{\mathbf{x}}_0 = \overline{\mathbf{x}}_{i-1}$$
(3.3)

$$\widehat{\Sigma}_{k+1} = \mathbf{F}_{\widetilde{\mathbf{x}}_k} \widehat{\Sigma}_k \mathbf{F}_{\widetilde{\mathbf{x}}_i}^T + \mathbf{F}_{\mathbf{w}_k} \mathbf{Q}_k \mathbf{F}_{\mathbf{w}_i}^T; \widehat{\Sigma}_0 = \overline{\Sigma}_{i-1}, \quad (3.4)$$

여기서 $Q_k \doteq w_k$ 의 공분산을 의미하며, $F_{\tilde{x}_k}$ 와 F_{w_k} 는 각각의 아래 첨자와 $(x_{k+1} \boxminus \hat{x}_{k+1})$ 의 자코비안을 의미한다. 또한 \hat{x}_{k-1} 역시 日를 이용해 구할 수 있는데, 해당 과정은 [14]에 소개되어 있다. B-spline interpolation 은 이산 모델의 정확도와 밀접한 관련이 있으며, 최적화 이전의 이산 모델이 정확하지 않기 때문에 LiDAR 측정값을 이용하여 업데이트를 한 후 ⊟ 연산자를 이용해 다시 한 번 계산하는 과정이 필요하다.

제 2 항 B-spline 보간을 이용한 연속 시간 궤적 추정

IMU 이산 모델의 전파 후, control points 라고 알려진 4 개의 global frame 의 행렬 ($\mathbf{T}_{k_{l}}^{k-1}$ to $\mathbf{T}_{k_{l}}^{k+2}$)을 활용해 B-spline interpolation 을 수행 할 수 있다 [16]. 해당 interpolation 을 통해 원하는 시간대의 경로를 추정이 가능 하기 때문에 이는 비동기 센서 시스템에서 매우 유용하며 시간적 보상이 가능하도록 만들어 줄 수 있다. 게다가, interpolation 의 smoothness 덕분에 곡률 및 연속적인 환경에서 큰 효과를 볼 수 있다. 이 때 B-spline Interpolation 은 IMU 이산 모델의 정확도에 따라 함께 정확도가 결정되므로, \mathbf{x}_{i-1} 이전 state 의 control points 는 \square 연산자를 활용해 다시 연산되어 진다. 위의 내용을 수식화 하면, t $\in [t_k, t_{k+1})$ 사이 의 변환 행렬은 아래와 같다.

$${}^{B}\mathbf{T}_{GI}(s(t)) = \mathbf{T}_{GI}^{k-1}\Pi_{n=1}^{3}\exp\left(\tilde{\mathbf{B}}_{j}(s(t))\Omega_{k+n-1}\right), \qquad (3.5)$$



그림 3: IMU inputs are advanced using a discrete model and then processed through B-spline interpolation. The LiDAR set that matches with the smallest time difference is searched. Based on the interpolation, (i) points from each LiDAR are undistorted by transformed into a single frame (the last point) and (ii) the temporal discrepancy is compensated by the relative transformation among the last points in each LiDAR.

이 때, $t_k = k\Delta t$, $\Delta t = t_{k+1} - t_k$ 이다. 또한, $s(t) = (t - t_k)/\Delta t$ 이며 Ω_k 는 incremental control pose 로 $\Omega_k = \log((\mathbf{T}_{GI}^{k-1})^{-1}\mathbf{T}_{GI}^k)$ 과 같이 계산된다. \tilde{B}_j 는 s 의 제곱항을 포함하는 행렬이며, 지금부터 ^BT 는 B-spline interpolation 을 통해 얻어진 변환행렬을 의미한다. 마지막으로, $\hat{\Sigma}_{k+1}$ 은 t $\in [t_k, t_{k+1})$ 에 얻어진 변환 행렬에 할당된다. 전체적인 과정은 [그림 3] 에 나타나 있다.

제 3 항 LiDAR 측정값 보정 및 다중 라이다 동기화

B-Spline Interpolation 에도 불구하고, Global frame 에서의 위치 변환행렬의 정확도는 LiDAR 간의 도착 시간 차이에 직접적으로 영향을 받는다. 따라서 도착 시간 차이의 합을 최소화하는 LiDAR 세트를 선택한다. 왜곡 보정을 위해, Secondary LiDAR 인 S 에 초점을 맞추며, 다른 LiDAR 에도 동일한 과정을 적용한다. 보정의 첫 단계는 다른 시간에 얻은 점들을 단일 frame 으로 병합하는 것이다. 이는 frame 간의 상대적 변환을 결정함으로써 이루어진다. 이전 연구에서는 추론을 위해 대략적인 IMU 이산 모델 또는 선형 보간을 사용하였다. 기존의 연구와 다르게 B-Spline interpolation 을 사용하면 시간 t_j 에서의 점 $p_{sj} = t_l$ 의 프레임으로 변환하여 왜곡이 보정된 점 $p_{sj}^u = 조금$ 더 정확하게 얻을 수 있다.

$$p_{Sj}^{u} = \mathbf{T}_{IS}^{-1B} \mathbf{T}_{I^{I}I^{j}} \mathbf{T}_{IS} p_{S^{j}}, \qquad (3.6)$$

여기에서 t_i 은 S 에서 가장 늦게 취득된 점의 시간이고, 왜곡이 보정된 점은 u 를 위 첨자로 사용하여 식별된다. 모든 LiDAR 는 측정 시간이 다르며, 이 시간 차이는 개별적으로 보정되어야 한다. 따라서, 이를 병합할 때 가장 최근의 LiDAR (P, [그림 3]에서 파란색 점으로 표시됨)의 가장 늦게 취득된 시간 t_i 를 활용하여 다른 LiDAR 에서 취득한 점들을 변환한다.

 $p_{P^{i}S^{j}} = \mathbf{T}_{IP}^{-1B} \mathbf{T}_{I^{i}I^{l}} \mathbf{T}_{IS} p_{S^{j}}^{u} = \mathbf{T}_{IP}^{-1B} \mathbf{T}_{I^{i}I^{j}} \mathbf{T}_{IS} p_{S^{j}}.$ (3.7)

이 변환은 frame 변경에 대한 왜곡 보정과 시간 보상을 모두 포함하며, 이러한 변경과 관련된 오류가 누적될 수 있다. 따라서 이를 반영해 줄 불확실성을 각 점 별로 전파할 수 있어야 한다.

제 3 절 점 별 라이다 불확실성 전파

이런 불확실성은 오류를 최적화에 포함시키기 위해 공분산을 사용하여 각 점에 전파되어야 한다. ^BT 와 T_{IL}의 공분산은 (3.4)와 Iterated Error State Kalman Filter를 통해 얻을 수 있다. 또한, 역행렬의 공분산은 Σ_{inv} =TΣT^T를 통해 계산된다. 여기서 T는 T⁻¹의 수반 행렬이며 변환 행렬과 공분산은 {T_{pisj},Σ_{pisj}}로 결합할 수 있다 [33]. Σ_{pisj}의 대각 성분은 취득 시간 불확실성을 나타내며, 이는 [그림 1-(b)]에 시각화 되어 있다.

이전 연구와 비교해 해당 논문은 세 가지 중요한 개선점을 보인다. 첫째, 각 LiDAR 점의 취득된 시간에 따라 불확실성을 구별하며, 이는 각 점과 관련된 불확실성 모델링을 보다 정확하게 하도록 한다. 이는 M-LOAM [12]에서 각 LiDAR 점들이 동일한 불확실성을 가진다고 가정한 것과는 대조적이다.

둘째, Primary LiDAR의 명시가 더 이상 필요하지 않다. M-LOAM에서는 Extrinsic Parameter에 해당하는 공분산이 Secondary

LiDAR에만 결합되므로, Secondary LiDAR의 공분산은 항상 Primary LiDAR보다 높다. 하지만 해당 논문에서 제시하는 방법은 Primary LiDAR를 명시적으로 지정하지 않아 보다 일반적이다. 또한, LiDAR와 IMU 사이의 해당 공분산을 활용하여, 모든 공분산이 Extrinsic Parameter 공분산과 결합됩니다. 이는 모든 공분산이 Extrinsic Parameter 공분산에 동일하게 영향을 받게 한다.

마지막으로, 이전 연구에서는 global frame에서 각 점들에 불확실성을 전파하지만, 해당 논문에서의 접근 방식은 각 점에 불확실성을 전파할 때 LiDAR 프레임으로 제한한다. 이와 같은 선택은 각 LiDAR에서 취득된 점을 통합할 때만 불확실성을 고려하기 위해서이다. 점을 frame P로 변환함으로써, 변환된 점은 다음과 같이 표현된다.

$$p_{P^{i}S^{j}} \triangleq \widehat{\mathbf{T}}_{P^{i}S^{j}}\widehat{p}_{S^{j}} = \exp\left(\xi_{P^{i}S^{j}}^{\wedge}\right)\mathbf{T}_{P^{i}S^{j}}\left(p_{S^{j}}+D\zeta\right) \quad (3.8)$$
$$\approx \left(I + \exp\left(\xi_{P^{i}S^{j}}^{\wedge}\right)\right)\mathbf{T}_{P^{i}S^{j}}\left(p_{S^{j}}+D\zeta\right)$$

여기서 ξ는 변환행렬의 오류이고 ζ∈ R³는 LiDAR 측정값의 변동 범위이다. 또한, (3.8)의 p는 스케일 값 1이 추가된 4 × 1 벡터이며, D는 0항이 추가된 3 × 1에서 4 × 1로 차원을 변환하는 확장 행렬이다. 2차항은 계산 비용이 많이 들지만 상대적으로 작은 효과를 가지므로, 1차항만 고려하여 **Î**p̂ ≈ q + Qθ로 표현할 수 있다.

$$\mathbf{q} := \mathbf{T}p, \ \mathbf{Q} := \begin{bmatrix} (\mathbf{T}p)^{\odot} & \mathbf{T}D \end{bmatrix}, \ \begin{bmatrix} \boldsymbol{\varepsilon} \\ \boldsymbol{\eta} \end{bmatrix}^{\odot} := \begin{bmatrix} \boldsymbol{\eta}\mathbf{1} & -\boldsymbol{\varepsilon}^{\wedge} \\ \mathbf{0}^{T} & \mathbf{0}^{T} \end{bmatrix}$$
(3.9)
$$\boldsymbol{\theta} := \begin{bmatrix} \boldsymbol{\xi}^{T} & \boldsymbol{\zeta}^{T} \end{bmatrix}^{T}, \ \boldsymbol{\theta} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Xi}), \ \boldsymbol{\Xi} = \operatorname{diag}(\boldsymbol{\Sigma}_{P^{I}S^{j}}, \mathbf{Z}),$$

LiDAR 측정값의 공분산은 Z 로 표기되었으며, *p_{pisi}* 가 가우시안 분포를 따르므로, 점의 불확실성은 (3.9)를 통해 Σ_p = QEQ^T로 얻을 수 있다. 결과적으로 얻어진 불확실성은 점 p의 범위와 관련된 (Tp)^O 와 획득 시간과 관련된 Σ_{pisi} 로부터의 구해진다. 시간에 따른 IMU 이산 모델 오류로 인한 모호성과 진동으로 인해 범위가 증가함에 따른 노이즈 효과를 이와 같이 점별 불확실성을 이용해 처리함으로써 향상된 정확도를 달성할 수 있다.

제 4 절 Kalman Filter 기반 위치 최적화

제 1 항 다중 LiDAR 시스템 전처리

제안된 방법은 지역적인 평면성 가정 하에 특징 추출 없이 평면측정 모델을 활용한다. LiDAR frame에서의 점들, {*p_si*,*j* = 1,...,1}은 다음 수식을 통해 global frame으로 변환될 수 있다:

$$\mathbf{p}_{GS^{j}} = \mathbf{T}_{GI^{i}} \mathbf{T}_{IP} \mathbf{p}_{P^{i}S^{j}} = \mathbf{T}_{GI^{i}} {}^{B} \mathbf{T}_{I^{i}I^{l}} \mathbf{T}_{IS} \mathbf{p}_{S^{j}}^{u}$$
(3.10)

마지막으로 스캔된 LiDAR P에 대해, (^BT_{Gl})^{-1B}T_{Gl}은 보상해야 할 시간 차이가 없으므로 단위 행렬과 동일하다. 평면을 근사화 하기 위해, 해당 논문은 ikd-Tree에서 측정점으로부터 가장 가까운 다섯 개의 이웃 점을 선택한다. 이 과정을 통해 시스템은 ikd-Tree에 포함된 점들의 관련 불확실성을 측정 모델에 통합한다. 일반성을 위해, LiDAR S 혹은 P를 모두 L로 사용한다. 점 p_Lj에 대한 평면의 가중 샘플 공분산 Σ_lj는 다음과 같이 계산된다:

$$\Sigma_{L^{j}} = \sum_{n=1}^{5} w_{n}^{2} \Sigma_{n}, \quad w_{n} = \frac{\tau - \operatorname{tr}(\Sigma_{n})}{\sum_{n=1}^{5} [\tau - \operatorname{tr}(\Sigma_{n})]}$$
(3.11)

여기서 τ는 불확실성 임계값을 나타내며, 이는 다음 절의 매핑 과정에서도 사용된다. 평면의 법선 벡터인 **v**^j_{GL}와 평면 공분산인 Σ_p를 기반으로, 측정 모델은 다음과 같이 계산된다:

$$\mathbf{0} = \mathbf{h}_{L^{j}}\left(\mathbf{x}_{i}, \mathbf{n}_{L^{j}}\right) = \frac{\mathbf{v}_{GL^{j}}^{T}\left(\mathbf{T}_{GI^{i}}{}^{B}\mathbf{T}_{I^{i}I^{l}}\mathbf{T}_{IL}\left(\mathbf{p}_{L^{j}}^{u} + \mathbf{n}_{L^{j}}\right) - \mathbf{q}_{GL^{j}}\right)}{\text{FIC}\left(\text{tr}\left(\Sigma_{L^{j}}\right), s_{max}, s_{min}\right)} \quad (3.12)$$

여기서, $\mathbf{n}_{L^{j}}$ 는 LiDAR에서의 노이즈를 나타내며, $\mathbf{q}_{GL^{j}}$ 는 평면 위에 위치한 점이다. 또한, **h**는 \mathbf{T}_{GI} 와 \mathbf{T}_{IL} 을 포함한 상태의 용어 요약인 측정 모델을 나타낸다. 여기서, Fixed Interval Conversion (FIC)를 이용하여 불확실성을 일정 범위 내로 제한하며, 다음과 같다.

$$FIC(V, I_{max}, I_{min}) = \frac{(I_{max} - I_{min})(V - V_{min})}{V_{max} - V_{min}} + I_{min}, \quad (3.13)$$

I_{max} 와*I_{min}* 은 재조정 구간이며, 마찬가지로 *V_{max}* 와 *V_{min}* 은 *V* 의 최대 및 최소 값이다. FIC를 활용함으로써, 설정된 범위 내에서 값들을 조정하여 과적합이나 특정 데이터를 무시하지 않는, 성능과 신뢰성을 조절하는 불확실성을 균형 잡는다.

제 2 항 사전 및 측정값 기반 위치 분포 및 오차

상태 추정은 [그림 2]와 같이 세 가지 요소로 구성되어 있으며, 이는 상태 전파, 잔차 계산, 그리고 상태 업데이트이다. 상태 전파 요소는 식 (3.3)과 (3.4)에서 표현되었으며, 이는 사전 분포로 활용되며, 그 오차 상태는 다음과 같이 얻어집니다.

 $\mathbf{x}_{i} \boxminus \widehat{\mathbf{x}}_{i} = \left(\widehat{\mathbf{x}}_{i}^{\kappa} \boxplus \widetilde{\mathbf{x}}_{i}^{\kappa}\right) \boxminus \widehat{\mathbf{x}}_{i} = \widehat{\mathbf{x}}_{i}^{\kappa} \boxminus \widehat{\mathbf{x}}_{i} + \mathbf{J}^{\kappa} \widetilde{\mathbf{x}}_{i}^{\kappa} \sim \mathcal{N}(\mathbf{0}, \widehat{\Sigma}_{i}) \quad (3.14)$

J^{κ}는 ($\hat{x}_{i}^{\kappa} \boxplus \tilde{x}_{i}^{\kappa}$) 曰 \hat{x}_{i} 가 $\tilde{x}_{i}^{\kappa} = 0$ 인 상태에서의 자코비안을 나타내며, $\kappa = 1$ 일 때, J^{κ} = I 이며 \hat{x}_{i} 의 변환행렬은 ^BT_{GIⁱ}가 된다. 마찬가지로 측정값 기반 위치 분포 및 오차는 1차식 근사를 활용하여 다음과 같이 표현된다.

$$\mathbf{0} = \mathbf{h}_{L^{j}}(\mathbf{x}_{i}, \mathbf{n}_{L^{j}}) \simeq \mathbf{h}_{L^{j}}(\widehat{\mathbf{x}}_{i}^{\kappa}, \mathbf{0}) + \mathbf{H}_{L^{j}}^{\kappa} \widetilde{\mathbf{x}}_{k}^{\kappa} + \mathbf{v}_{L^{j}} \quad (3.15)$$

$$-\mathbf{v}_{j} = \mathbf{z}_{L^{j}}^{\kappa} + \mathbf{H}_{L^{j}}^{\kappa} \widetilde{\mathbf{x}}_{k}^{\kappa} \sim \mathcal{N}(\mathbf{0}, \Sigma_{L^{j}})$$

이 때, $\mathbf{H}_{L^{j}}^{\kappa} 는 \mathbf{h}_{L^{j}}(\hat{\mathbf{x}}_{i}^{\kappa} \boxplus \tilde{\mathbf{x}}_{i}^{\kappa}, \mathbf{n}_{L^{j}})$ 와 $\tilde{\mathbf{x}}_{i}^{\kappa}$ 의 자코비안을 나타내며, $\boldsymbol{v}_{L^{j}}$ 는 (3.9)에서 계산된 불확실성을 포함한다.



그림 4: Diagram for Localization weight. In Case 1, it's better to depend on the IMU instead of the LiDAR measurements since there are only measurements in y axis. Conversely, in scenarios like Case 2, where measurements are available from all directions, it's more beneficial to rely on the LiDAR rather than the IMU. This weight is effective in preventing failures in degenerate environments.

제 3 항 위치 추정을 위한 Localization 가중치

Localization 가중치는 [그림 4]와 같이 좁은 환경 등에서 IMU를 통해 구해진 사전 분포와, LiDAR를 통해 구해진 측정값 분포에서 어떤 분포에 가중치를 줄지 결정할 수 있는 값이다. [그림 4]에서 Case 1의 경우 한 축의 측정값만 존재하므로 IMU를 좀 더 의존하는 것이 좋으며, 반대로 Case 2의 경우 LiDAR 측정값이 모든 방향에 존재하므로 LiDAR 측정값을 사용하는 것이 더 효과적이다. 따라서, 주변의 환경을 인식하여 이와 같은 가중치를 구할 수 있어야 한다. 4절 1항에서 이미 주변 환경에서의 법선 벡터를 추출하였으므로, 법선 벡터의 분포를 바탕으로 측정값이 모든 차원에 고르게 분포되어 있는지 확인할 수 있다. 따라서, Localization 가중치는 법선 벡터 행렬의 Singular Value Decomposition에서 얻은 최대값 σ₁과 최솟값 σ₃의 비율을 취함으로써 결정할 수 있다.

제 4 항 전체 시스템 최적화

최종적으로 (3.14)와 (3.15)의 분포를 이용해, state estimation problem을 maximum a posteriori (MAP) 문제로 바꾼다면 아래와 같다.

$$\min_{\widetilde{\mathbf{x}}_{i}^{\kappa}} \left(\|\mathbf{x}_{i} \boxminus \widehat{\mathbf{x}}_{i}\|_{\widehat{\Sigma}_{i}}^{2} + w_{l}^{2} \sum_{L=P,S} \sum_{j=1}^{m} \|\mathbf{z}_{L^{j}}^{\kappa} + \mathbf{H}_{L^{j}}^{\kappa} \widetilde{\mathbf{x}}_{i}^{\kappa}\|_{\mathbf{R}_{L^{j}}}^{2} \right)$$
(3.16)

 $\mathbf{R}_{L^{j}}$ 는 FIC(tr(Σ_{L^{j}}), \mathbf{R}_{max} , \mathbf{R}_{min})의 결과이다. Localization weight (w_{l})은 측정 모델 분포에 가중치로 주어진다. 만약 w_{l} 이 b_{max} , b_{min} 의 경계 밖에 있다면, l_{max} , l_{min} 의 값이 각각 w_{l} 에 할당된다. 그렇지 않으면 w_{l} 의 값은 $w_{l} = FIC(w, l_{max}, l_{min})$ 을 이용해 얻을 수 있다. 그 후, 칼만 필터를 활용하여 (3.16)의 문제를 해결할 수 있다

$$\mathbf{K} = \left(\mathbf{H}^{T}\mathbf{R}^{-1}\mathbf{H} + \mathbf{P}^{-1}\right)^{-1}\mathbf{H}^{T}\mathbf{R}^{-1}, \qquad (3.17)$$
$$\widehat{\mathbf{x}}_{i}^{\kappa+1} = \widehat{\mathbf{x}}_{i}^{\kappa} \boxplus \left(-\mathbf{K}\mathbf{z}_{i}^{\kappa} - (\mathbf{I} - \mathbf{K}\mathbf{H})\left(\mathbf{J}^{\kappa}\right)^{-1}\left(\widehat{\mathbf{x}}_{i}^{\kappa} \boxminus \widehat{\mathbf{x}}_{i}\right)\right)$$

$$\begin{split} \mathbf{H} &= w_l \times \left[\mathbf{H}_{S^1}^{\kappa^T}, \cdots, \mathbf{H}_{S^m}^{\kappa^T}, \mathbf{H}_{P^1}^{\kappa^T}, \cdots, \mathbf{H}_{P^m}^{\kappa^T}\right]^T, \mathbf{R} = \operatorname{diag}(\mathbf{R}_{S^1}, \cdots, \mathbf{R}_{S^m}, \mathbf{R}_{P^1}, \\ \cdots, \mathbf{R}_{P^m}), \mathbf{P} &= (\mathbf{J}^{\kappa})^{-1} \widehat{\sum}_i (\mathbf{J}^{\kappa})^{-T}, \mathbf{z}_i^{\kappa} = w_l \times \left[\mathbf{z}_{S^1}^{\kappa^T}, \cdots, \mathbf{z}_{S^m}^{\kappa^T}, \mathbf{z}_{P^1}^{\kappa^T}, \cdots, \mathbf{z}_{P^m}^{\kappa^T}\right]^T \stackrel{\text{def}}{=} \\ \text{의미하며}, \|\widehat{\mathbf{X}}_i^{\kappa+1} \boxminus \widehat{\mathbf{X}}_i^{\kappa}\| < \epsilon \stackrel{\text{def}}{=} \mathbb{V} \stackrel{\text{def}}{=} \mathbb{V} \stackrel{\text{def}}{=} \mathbb{V} \text{def} \text{def}. \end{split}$$

추정 값과 이에 해당하는 공분산은 아래와 같다.

$$\bar{\mathbf{x}}_i = \widehat{\mathbf{x}}_i^{\kappa+1}, \quad \bar{\boldsymbol{\Sigma}} = (\mathbf{I} - \mathbf{K}\mathbf{H})\,\mathbf{P} \tag{3.18}$$

추가로 기존의 Kalman Gain (K)의 경우 PH^T(HPH^T + R)⁻¹ 로 계산되며, HPH^T + R의 차원이 측정값의 수와 동일하기 때문에 역행렬을 구하는 과정에서 매우 많은 계산이 발생한다. 이런 Kalman Filter는 측정값에 해당하는 법선 벡터가 적은 상황에서 이점이 있으나, 다중 LiDAR 시스템에서는 측정값이 일반적으로 많기 때문에 이는 훨씬 더 많은 계산을 요구하며 실시간 상태 추정을 불가능하게 만들 수 있다. 하지만 선행 연구 [23]에서 K = (H^TR⁻¹H + P⁻¹)⁻¹H^TR⁻¹ 역시도 같은 결과가 도출된다는 것이 증명되었으며, 역행렬 계산 시 측정값의 차원이 아닌 State의 차원에 해당하는 행렬을 계산하기 때문에 계산이 매우 줄어들어 실시간 상태 추정을 유지할 수 있다. R 에 해당하는 차원은 측정값의 차원과 같지만, 이는 대각행렬이기 때문에 단순히 역수를 구한다면 역행렬을 구할 수 있게 되므로 계산에 전혀 문제가 되지 않는다.

제 5 절 점 별 불확실성을 고려한 맵핑

각 점의 불확실성을 효과적으로 관리하기 위해, 지도에 점을 저장하기 전 점의 불확실성을 평가한다. 만약 각 점의 불확실성, tr(Σ_L) 이 미리 정의된 τ보다 크면, 해당하는 점은 데이터베이스에 저장되지 않는다. 반대로 조건에 부합된다면 점들은 다운샘플링을 통해 ikd-Tree에 저장된다. 본래의 ikd-Tree와는 달리, 해당 논문에서는 이러한 저장 과정에서 불확실성을 고려한다. 기존의 다운샘플링 과정은 정확한 맵핑을 위해 트리 해상도의 중심에 가까운 포인트만을 유지하도록 설계되었지만, 제안한 방법은 만약 삽입 포인트가 트리의 중심에서 0.05m 이내에 위치한다면 트리 내에서 낮은 불확실성을 가진 포인트가 트리에 유지된다.

Dataset	Number	Lidar	IMU	Environment
Hilti	1 2	OS0-64 Livox Horizon	200Hz	Indoor & Outdoor
UrbanNav	1 2 3	HDL-32E VLP-16 LS-16C	400Hz	Urban with Skyscrapper
Ours	1 2 3	OS2-128 Livox Avia Livox Tele	100Hz	Urban with Challenges

표 1: Dataset Description

제 1 절 실험 환경 소개

제 1 항 데이터셋 소개

제시한 방법론의 성능을 다양한 환경에서 평가하기 위해, 해당 논문 은 Hilti SLAM Dataset 2021 [1], UrbanNav[2], 그리고 직접 취득한 City 데이터셋[34]에 대해서 실험을 수행하였다. Hilti 데이터셋은 소규 모 실내 및 실외 환경에서 휴대용 시스템을 사용했을 때 해당 방법론이 효과가 있는지 평가한다. UrbanNav 데이터셋은 넓은 도시 환경 및 차량 시스템에서 해당 방법론의 성능을 평가하기 위해 활용된다. 오픈 데이터 셋 외에도, 해당 논문은 U턴과 터널을 포함하여 높은 속도(약 50 km/h) 와 같은 도전적인 환경에서의 성능을 평가하기 위해 데이터셋을 수집하 였다 (City Dataset). 수집 과정에서 각 센서의 시간 프레임을 동기화하 기 위해, PTP가 사용되었다. 그러나 이것은 모든 센서가 동시에 데이터 를 취득함을 의미하지 않고 Universal Time Coordinate (UTC)에서 각 센서가 데이터를 취득함을 의미한다. 데이터셋에 사용한 센서 및 각 특 징은 [표 1]에 나열되어 있으며, 각 시퀀스에 대한 자세한 설명은 후속 섹션에서 추가로 제공된다.

제 2 항 평가 방법 소개

해당 논문은 Fast-LIO2 (단일 LiDAR) [14], M-LOAM [12] (다중 LiDAR), 그리고 LOCUS 2.0 [4] (다중 LiDAR)를 포함한 세 가지 최신 기법과 비교한다. 공정한 비교를 보장하기 위해, 다음과 같은 전략을 사 용하였다. Fast-LIO2는 단일 LiDAR만을 지원하기 때문에, 가장 많은 점들을 제공하는 중앙 LiDAR를 인풋으로 활용하였다. Hilti 데이터셋에 대해서는 각 LiDAR에 해당하는 궤적을 모두 얻었으며, OSO-64와 Livox Horizon에 대해 FAST-O와 H로 표시하여 구분하였다. M-LOAM은 회전 LiDAR만을 지원하기 때문에, Livox LiDAR의 점들을 평 면 특징점이라 가정하고 입력 값으로 주었다. LOCUS 2.0에 대해서는 GICP의 매개변수를 특정 환경에 맞게 조정하였다. 또한, 선택적으로 오 도메트리를 입력 값으로 제공할 수 있는데, 각 데이터셋에는 오도메트리 가 따로 존재하지 않기 때문에 입력 값으로 사용하지 않았다. 사용한 매 개변수는 **Z** = diag(0.05, 0.05, 0.05), (*s_{min}, s_{max}*) = (1, 1.25), (*b_{min}, b_{max}*) = (0.2, 0.8), (*l_{min}, l_{max}*) = (0.5, 3), (*R_{min}, R_{max}*) = (0.0075, 0.0125), τ = 1 와 같으며, 각 데이터셋에 따라 약간의 변동이 있었다.

방법들의 성능을 정량적으로 비교하기 위해, Evo evaluator [35]를 이용하여 ATE와 RTE의 RMSE를 계산하였다. ATE는 meter와 degree로, RTE는 퍼센트와 deg/meter로 측정되었다. Hilti 데이터셋에 대해서는 데이터셋에 제공된 evaluator를 활용하여 ATE_t, ATE의 translation 요소를 계산하였다. City Dataset에서는 Inertial Navigation System (INS)를 이용하여 Ground-truth가 얻어진다. 결과의 신뢰성을 보장하기 위해, INS_SOLUTION_FREE라는 상태를 가진 값들을 활용하였다.

	Ours	Fast-H	Fast-O	M-LOAM	LOCUS 2.0
Basement	0.036	0.709	0.046	0.115	0.120
Campus	0.046	0.063	0.063	0.386	0.087
Construct	0.063	0.200	0.088	2.647	0.290
LAB	0.024	Err	0.026	0.064	0.040
UZH	0.177	0.233	0.184	0.276	0.177

표 2: ATE_t for Hilti SLAM Dataset 2021

The best results are in **bold** and the second-best results are in *italic*.



그림 5: (a) The accumulated scans from LAB are displayed. The Ouster (gray) is contrasted with Livox (red), showcasing a limited FOV, leading to localization failures due to inadequate features. (b-c) These illustrate maps with the Fast-O and ours, using the Parking dataset. In the red box, the Fast-O cannot align the map upon returning to the start point, however, our result is well-aligned.

제 2 절 실험 결과 분석

제 1 항 Hilti 2021 SLAM Dataset

Hilti evaluator로부터 얻은 결과는 [표 2]에 제시되어 있다. 모든 방법 중에서, 해당 논문은 모든 시퀀스에서 가장 정확한 결과를 제공한다. 흥미롭게, Fast-O는 다중 LiDAR를 사용하지 않음에도 불구하고 대부분의 시퀀스에서 두 번째로 좋은 성능을 달성하였다. M-LOAM과 LOCUS 2.0은 다른 알고리즘에 비해 정확도가 떨어지며, 이는 LiDAR 간의 비동기화에 기인합니다. LAB와 UZH의 경우에는 실내 환경에서 강한 움직임이 없기 때문에 비동기성의 영향이 최소화되어 큰 에러가 도출되지 않았다. 그러나, 더욱 강한 움직임이 포함된 Campus와 Construct의 경우, 센서들의 고유한 비동기성 때문에 오류가 더욱 두드러진다. 이러한 도전에도 불구하고, 해당 논문은 B-Spline Interpolation의 시간적 보상과 불확실성의 고려 덕분에 모든 시퀀스에서 견고성을 보여준다.

한편, Fast-O와 달리, Fast-H는 LiDAR의 제한된 FOV 때문에 상당히 저하된다. Livox LiDAR의 제한된 FOV는 [그림 5(a)]에서 보여지는 것처럼 시야각의 제한에 노출될 가능성이 높다. 이 FOV에 의한 제한은 풍부한 특징점이 존재하는 경우에도 상당한 궤적 오차를 초래할 수 있다. 이 관점에서, Fast-O는 대부분의 환경에 대한 최선의 선택으로 보이지만, 특히 Parking의 경우에는 최적의 해결책이 아니었다. Parking의 경우 Ground-truth가 제공되지 않기 때문에, [그림 1(b)]에서 보여지는 것처럼 완성된 지도를 이용해 정성적 평가를 진행하였다. Parking의 중앙에서는 상당수의 점들이 LiDAR에 감지되지 않아 지도에서 큰 오차를 초래하여, 단순히 Fast-O만으로는 정확한 지도 작성을 할 수 있지 않았다. 반면에, 제안한 방법은 [그림 1(c)]에서 보여지는 것처럼 추가적인 LiDAR를 통합함으로써 이러한 문제를 완화할 수 있으며, 다중 LiDAR 시스템의 견고성을 보여주었다.

		Fast-LIO2	M-LOAM	LOCUS 2.0	Ours
Mongok	$\begin{array}{c} \text{ATE}_t\\ \text{ATE}_r\\ \text{RTE}_t\\ \text{RTE}_r \end{array}$	5.917 4.039 0.188 0.749	25.899 9.140 0.632 1.006	6.846 5.616 <i>0.174</i> 0.710	2.579 2.383 0.167 0.736
Whampoa	$\begin{array}{c} \text{ATE}_t \\ \text{ATE}_r \\ \text{RTE}_t \\ \text{RTE}_r \end{array}$	7.066 7.066 0.390 1.034	31.482 8.286 0.710 1.213	18.124 9.404 <i>0.339</i> 1.238	4.236 4.600 0.207 1.033
TST	$\begin{array}{c} \text{ATE}_t \\ \text{ATE}_r \\ \text{RTE}_t \\ \text{RTE}_r \end{array}$	8.783 6.640 0.494 1.264	53.682 21.584 2.177 1.355	33.292 13.367 0.841 1.748	2.342 5.085 0.351 1.261

표 3: UrbanNav Dataset Evaluation

제 2 항 UrbanNav Dataset

UrbanNav 데이터셋은 두 개의 회전 LiDAR가 기울어진 상태로 차에 장착되어 있으며, 중앙에 하나의 회전 LiDAR를 두어, 총 세 개의 회전 LiDAR로 구성되어 있다. 해당 구성은 스캔 영역을 최대화하면서 최소한의 중복된 영역을 만들어, 중앙 LiDAR로는 보이지 않는 고층 건물 등을 감지할 수 있다. 그러나, 이 데이터셋은 앞선 Hilti 데이터셋에 비해 더 높은 속도로 데이터를 취득하여 비동기로 인한 LiDAR의 시간적 차이에 해당하는 위치 변동이 큰 편이며, 많은 수의 동적 객체를 포함하고 있으므로 정확한 위치 추정을 하는데 어려움을 수반하고 있다.

[표 3]에서 보여지는 것처럼, 해당 논문에서 제안한 방법은 다른 방법들을 능가합니다. Mongok에서, 대부분의 방법들은 단일 루프의 반복적인 통과로 인해 적은 오류를 보여준다. 그러나, Whampoa에서는 경로 중간에 존재하는 터널로 인해 대부분의 알고리즘에서 더욱 큰 오류를 보여준다 [그림 10(a)]. 또한, TST에서의 차량 속도는 Mongok 시나리오의 약 두 배일 정도로 무척 빠른 편인데 이로 인해 앞서 언급되었던 비동기 시스템에서의 정합의 어려움으로 인해 다중 LiDAR 시스템의 오류를 무척 증가시켰다. 이러한 어려움에도 불구하고, 제안된 방법은 LiDAR 간의 상대적 변환을 정확히 추정하고 단일 LiDAR로는 불가능한 광범위한 스캔을 가능하게 하여 다른 에러들과 비교하였을 때 효과적으로 에러를 줄일 수 있었다. [그림 6, 그림 7].



그림 6: Trajectory of TST and Mongok. Our method (blue) demonstrates a highly close alignment with the Ground-truth (black).



그림 7: These images show the partial map from our method. Even points are arrived at different time, they are integrated accurately.

		Fast-LIO2	M-LOAM	LOCUS 2.0	Ours
City01	$\begin{array}{c} \text{ATE}_t\\ \text{ATE}_r\\ \text{RTE}_t\\ \text{RTE}_r \end{array}$	9.970 4.575 0.292 0.898	33.907 8.792 0.955 1.020	23.998 5.521 0.609 0.895	6.538 3.491 0.266 0.874
City02	$\begin{array}{c} \text{ATE}_t \\ \text{ATE}_r \\ \text{RTE}_t \\ \text{RTE}_r \end{array}$	35.308 7.473 0.608 1.179	72.382 4.683 3.665 1.104	58.211 4.722 1.531 1.167	6.707 3.522 0.565 1.084
City03	$\begin{array}{c} \text{ATE}_t \\ \text{ATE}_r \\ \text{RTE}_t \\ \text{RTE}_r \end{array}$	6.951 4.194 0.996 1.088	33.801 6.657 1.310 1.070	21.753 4.773 1.159 1.089	5.470 3.522 0.565 1.084

표 4: City Dataset Evaluation

제 3 항 Citty Dataset

직접 취득한 데이터셋인 CityO1-03은 각 시퀀스 별로 일련의 도전 과제를 제시하며, 표 4에서 보여지는 바와 같이 모든 시퀀스는 다른 데 이터셋에 비해 큰 에러를 가진다. 그럼에도 제안한 방법은 대부분의 경 우에서 우수한 성능을 달성하는 것을 볼 수 있다. CityO1은 많은 회전과 U-턴을 포함하고 있는데 U-턴을 완료한 후, M-LOAM과 LOCUS 2.0 은 이전에 생성된 맵과 정합하는 과정에서 시스템의 위치를 찾는데 실패 한다. 이러한 에러는 LOCUS 2.0에서 초기 추정치의 부재와 비동기 LiDAR 정합 능력의 부재로 기인한다. 결과적으로, 시스템은 위치 결정 을 위해 기존의 맵에 정합하는 것이 아닌, 추가적인 맵을 구축하는데 의 존하게 된다.

City02는 약 400m의 길이의 터널 환경을 특징으로 한다. 해당 시퀀스에서, 각 알고리즘은 터널 내의 점들 간에 대응관계를 찾지 못함으로써 위치 추정에 실패하여 상당한 오류를 발생시킨다.

마지막으로, CityO3는 총 4.3km의 경로를 포함하며, 많은 동적 객체가 존재한다. 또한, 출발점으로 돌아올 때까지 겹치는 경로가 없기 때문에 오류가 점차 누적될 수 있다. 제안한 방법은 경로 중간에 루프가 없음에도 불구하고 CityO3에서 낮은 오류를 보여주며, [그림 8(a)]에서는 한 바퀴를 돌고 난 후에도 맵은 잘 정렬되어 있음을 볼 수 있다, 또한 UrbanNav 데이터셋과 마찬가지로 비동기 LiDAR를 사용하더라도, 제안한 방법은 LiDAR 측정값 사이의 시간 보상과 불확실성 고려 덕분에 에러를 줄여 [그림 8(b)]과 같은 명확한 지도를 생성할 수 있었다.



그림 8: (a) The City03 map displayed here uses color-coding to indicate point height, transitioning from low (red) to high (green) values. On the right, the stacked scans from (b) our proposed method and (c) LOCUS 2.0 are demonstrated. Our method successfully corrects temporal discrepancies, resulting in reduced noise even when dealing with street markers, as highlighted in the red box.

제 3 절 모듈 별 기여도 분석

제 1 항 모듈 별 정확도 분석

전체 알고리즘 (FULL)은 B-Spline Interpolation (CNT)과 위치 결정 가중치와 함께하는 불확실성 전파 (UNC)로 구성되어 있다. 해당 모듈이 실제로 정확도에 기여하는 바를 파악하기 위해, IMU 이산 모델과 모든 점에 대한 균일 가중치를 사용하는 방법을 기준 (RAW로 표기)으로 하여 각 방법론과 비교한다. 또한, 우리는 점 별 불확실성 전파의 효과성을 확인하기 위해 스캔의 마지막 공분산을 사용하는 M-LOAM의 경우처럼 단일 상태 공분산만을 사용하는 방법과도 비교를 진행한다 (F-UNC). 모든 방법들은 ATE_t를 사용하여 검증된다. 또한, 확실한 차이를 확인하기 위해 에러가 큰 UrbanNav 데이터셋과 City 데이터셋을 이용해 확인하였다.

	RAW	CNT	F-UNC	UNC	FULL
City01	7.345	7.280	7.001	6.831	6.538
City02	7.346	6.835	7.058	6.844	6.707
City03	6.043	5.969	6.547	5.837	5.470
Mongok	2.652	2.597	2.645	2.611	2.579
Whampoa	4.728	4.463	4.657	4.078	4.236
TST	2.721	2.143	2.773	2.752	2.324

표 5: Component-wise comparison using ATE_t

[표 5]는 각 테스트 케이스에 대한 ATE_t 를 제시한다. RAW와 CNT는 모두 보간 기법을 활용하여 다중 LiDAR의 시간적인 불일치를 처리하지만, CNT는 B-Spline Interpolation을 사용함으로써 약간의 개선을 보여준다. B-Spline Interpolation은 특히 곡률이 높은 궤적을 보이는 데이터셋에 효과적이다. 예상대로, UrbanNav 데이터셋에서 ATE_t가 줄어들었는데, 해당 데이터셋은 City 데이터셋 (100Hz)보다 더 높은 IMU 주파수 (400Hz)를 가지고 있다. 이 결과는 B-Spline Interpolation을 통한 왜곡 해소 및 비동기 LiDAR의 시간 보상이 IMU 주파수에 관계없이 두 데이터셋 모두에서 효과적임을 보여준다. 또한, UNC에서 볼 수 있듯이, 점별 불확실성은 모듈 중 가장 에러를 크게 줄였으며, F-UNC 역시 일정 수준의 에러 감소를 보여주지만 제안한 방법만큼 줄이지는 못했다.

City01-03에서, FULL은 두 방법의 장점을 모두 효과적으로 활용하여 최상의 결과를 보였으나 UrbanNav 데이터셋은 환경 요인으로 인해 다양한 결과를 보여주었는데, 정지 상태와 저속 구간은 B-Spline Interpolation과 점 별 불확실성의 효과를 약하게 만들어 Mongok에서 대부분의 알고리즘이 비슷한 성능을 보여주었다. TST에서는 기울어진 LiDAR가 종종 더 적은 점을 감지하고 중앙 LiDAR에 비해 불확실성이 낮아졌기 때문에 UNC의 성능이 RAW에 비해 약간 떨어졌다, 이는 다시 FULL의 결과에 영향을 미쳤다. 반면에, Whampoa에서는 증가된 환경 복잡성이 주요 LiDAR가 변경되었을 때에도 점 별 불확실성과 위치 결정 가중치를 더 효과적으로 사용하게 하여 UNC의 성능을 크게 향상시켰다. 비록 B-Spline Interpolation과 점 별 불확실성 사이의 효과가 완전히 상호 보완적이지 않아 UNC에 비해 FULL의 성능을 떨어뜨리지만, 전체적으로 보았을 때 에러가 RAW에 비해 크게 줄어들었다. 결론적으로, 제안된 두 방법은 개별적으로 또는 결합하여 사용할 때 일반적으로 성능을 개선하는 것을 확인하였다.



(a) The Output of RAW

(b) The Output of UNC

그림 9: (a) This map is generated using Livox Horizon after the successful initialization (60sec from starting) in the LAB. (a) unsuccessful mapping in this degenerate environment. (b) consistent map by the proposed method overcoming the environment.

제 2 항 지도 작성 기여 효과

[그림 9]는 LAB 시퀀스에서 RAW와 UNC를 정성적 비교 결과를 보여준다. 좁은 실내 환경에서는 FOV가 매우 제한적이므로, 시스템이 Livox Horizon만으로 구성된 경우 정확한 위치 추정이 힘들 수 있다. [그림 9(a)]에서 볼 수 있듯, RAW는 일관적으로 위치 추정 및 지도 작성에 실패하나, UNC는 제한된 공간에서 성공적인 위치 추정을 해내며. 지도 작성도 성공적으로 완료하였다. [표 5]와 [그림 9]의 결과를 통해, 불확실성 모델이 제안된 방법에서 성능 향상에서 주요한 역할을 하는 것으로 추론할 수 있었다.



그림 10: Partial maps of (a) City02 and (b) Whampoa are shown, with points color-coded based on the localization weight, spanning from low (red) to high (pink). In both maps, there is a clear decrease in weight within the tunnel areas. However, the decrease is less pronounced in (b) because the top of the underpass is open and being scanned by an angled LiDAR.

제 3 항 Localization 가중치 기여 효과

[그림 10]에서는 City02와 Whampoa의 부분 지도를 보여준다. 터널과 지하도 환경 등에서 나타난 좁은 공간은 각 LiDAR의 점간 대응 오류를 유발시킬 수 있기 때문에, 사전 분포가 측정값 모델의 분포보다 더 큰 영향을 미치도록 계획하여야 한다. [그림 10]에서 볼 수 있듯이, Localization 가중치는 언급되었던 환경에서만 뚜렷하게 감소하는데, 이는 앞서 예상했던 결과와 동일하며, 도전적인 환경에서 Localization 가중치의 효과가 실제로 영향이 있음을 설명한다.

Whampoa City01						
LiDAR #	1	2	3	1	2	3
Point (Down.) #	4045	5647	6784	8554	10172	12244
LiDAR	HDL-32E	+VLP-16	+LS-C16	OS2-128	+Avia	+Tele
Preprocess	1.1	1.4	1.9	1.6	2.0	2.1
Pre-integration	2.9	4	6.4	2.6	3.5	4.9
B-Spline	7.4	8.6	9.4	7.9	10.5	13.1
Uncertainty	1.6	2.3	3.5	2.9	3.3	5
Kalman Filter	13.1	18.9	22.2	27.2	31.6	47.2
Mapping	2.0	3.4	4.3	4.2	5.5	6.7
Total	28.1	38.6	47.7	46.4	56.4	79

표 6: Time analysis according to the number of LiDAR [ms]. The numbering follows the description in [표 1].

*Specification: Intel i7 CPU@2.50Ghz and 48GB RAM



그림 11: The computation times relative to the number of LiDARs are displayed. Pre-process refers to the time taken to reformat the point cloud. Pre-integration involves the integration of IMU measurements and the undistortion of points. The usage of B-spline during undistortion is separately listed from Pre-integration. Uncertainty includes the time taken to calculate the point-wise uncertainty and the localization weight.

제 4 절 다중 라이다 개수 분석

제 1 항 시간 복잡도 분석

UrbanNav와 City 데이터셋 중 가장 거리가 긴 Whampoa와 CityO1에 대해 각 모듈별로 시간이 얼마나 걸렸는지를 분석하였다. 특히, 실시간 성능을 평가하기 위해, 각 모듈에 대한 스캔 당 평균 처리 시간을 측정하며, 정량적인 값은 [그림 11]과 [표 6에서 찾을 수 있다. 해당 실험을 진행할 때, 각 LiDAR 점군은 0.4m 해상도로 다운샘플링이 처리되었다. [표 6]은 다운샘플링 후에 남아있는 점의 수를 보여준다. 결과에서 알 수 있듯, 제안한 방법은 빠른 속도로 처리가 가능하며, 심지어 다중 LiDAR 시스템에 대해서도 실시간 성능을 완전히 지원한다. 또한, 제안한 모듈 중 하나인 불확실성 전파는, 성능을 크게 향상시킴에도 불구하고 최대 5ms 내에 완료될 수 있었다. 반면에, B-Spline Interpolation은 왜곡 보정이 다운샘플링 전에 이루어지기 때문에 모든 점에 대해 처리되어서 더욱 많은 시간을 소요하였다. 그러나, 많은 점을 처리해야 하는 CityO1에서도, 제안한 방법은 10Hz를 완전히 지원하는데 79ms를 소비하여 실시간성이 검증되었으며, LiDAR의 개수가 더욱 증가한 경우 왜곡보정 전에 다운샘플링을 수행함으로써 추가 성능 향상이 가능하다.

제 2 항 정확도 분석

LiDAR의 수가 정확도에 미치는 영향은 TST, Whampoa, City03, 총 세 가지 데이터셋을 사용하여 더욱 평가되었다. City03에서 RAW와 의 결과를 비교하면, LiDAR의 수가 증가함에 따라 오류가 감소하며, 사 분위 범위와 표준 편차를 조사할 때 오류 범위도 감소하는 것을 [그림 12(a)]에서 볼 수 있었다.



그림 12: (a) an evaluation for three cases for two datasets. The median values of ATE_t and standard deviations are shown as: (3.69, 1.22), (2.44, 0.96), (2.14, 0.98), (5.69, 2.53), (4.01, 2.69) and (3.58, 1.93). (b) represents the median ATE_t in City03 as a function of the number of LiDARs for the FULL and RAW.

특히, LiDAR의 수를 하나에서 두 개로 늘릴 때 오류 감소 효과가 더욱 컸으며, 두 개에서 세 개로 늘릴 때는 감소는 하였으나 그 정도가 약해졌다. 이에 대한 이유로는 두 개의 LiDAR로 늘림으로써 이미 충분한 제약 조건이 제공되기 때문이라고 추론되며, [그림 12(b)]의 표에서 볼 수 있듯이, 더 많은 LiDAR가 반드시 성능을 향상시킨다고 주장하는 것은 잘못된 생각일 수 있다. City03에서, 두 번째 (Livox Avia)와 세 번째 (Livox Tele) LiDAR를 추가하였다고 해서 오류를 효과적으로 줄이지 못했는데, 해당 LiDAR들의 좁은 FOV와 Ouster LiDAR와의 많은 겹침으로 인해 오류를 효과적으로 줄이지 못했다. 이것은 다중 LiDAR 시스템에서 LiDAR 배치의 중요성을 강조한다. 더욱이, 두 개의 LiDAR를 FULL로 사용하는 것은 세 개의 LiDAR를 RAW로 사용하는 것보다 우수한 성능을 나타냈는데, 이는 단순히 겹치는 영역에서 각 LiDAR 점군을 병합하는 것은 큰 성능 개선을 가져오지 않을 수 있다는 사실을 보여준다. 대신, B-Spline Interpolation을 사용하여 점들을 정확한 위치로 변환시키고, 각 점에 불확실성을 할당함으로써, 세 개의 LiDAR를 가진 FULL은 최상의 성능을 가질 수 있었다.

제 5 장 결론

제 1 절 요약

해당 논문에서는 비동기 다중 LiDAR-IMU 시스템에 대한 프레임워크를 제안했다. 제안된 프레임워크는 IMU 이산 모델을 기반으로 B-spline Interpolation를 활용하여 다중 LiDAR들 사이의 시간 차이를 완화하는데 사용된다. 비동기 시간 보상에서 일반적으로 발생하는 프레임 변환 중 누적된 불확실성을 완화하기 위해, 해당 논문에서는 IMU 취득 시간과 센서로부터의 취득된 점 거리를 기반으로 점별 불확실성을 전파하는 방법을 제안했다. 또한, 해당 논문은 좁은 통로나 터널 등과 같은 도전적인 환경에서 위치 추정 성능을 개선하기 위해 Localization 가중치를 제안하였다. 제안된 방법은 오픈 데이터셋 및 직접 취득한 City 데이터셋에서 광범위한 실험을 통해 검증되었으며, 정확도와 견고성 면에서 State-of-the-arts을 뛰어넘는 동시에 각 모듈들이 실제로 의미가 있는지 추가 실험을 통해 증명하였다. 또한, LiDAR의 개수에 따른 추가 실험을 진행하여 실시간 성능 및 개수에 따른 정확도 증가를 확인하였고, 해당 결과를 바탕으로 다중 LiDAR 시스템에서 LiDAR 배치의 중요성을 강조하였다.

제 2 절 향후 계획

해당 절에서는 향후에 추가적으로 수행할 수 있는 연구에 대한 계획을 의논한다.

- LiDAR 개수에 따른 배치에 관한 연구가 필요하다. 다중 LiDAR의 배치에 대한 연구 [36]는 존재하지만, 해당 연구는 어떻게 하면 효과적으로 넓은 범위를 커버할지에 집중되어 있다. 하지만 특정 환경에서는 높은 고도에 물체가 위치해 있지 않아 LiDAR를 기울여서 설치할 필요가 없을 수 있다. 또한, 선행 연구는 회전 LiDAR에만 집중되어 있지만, 최근에는 Solid State LiDAR 등의 등장으로 좁은 FOV LiDAR에 대한 수요도 무시할 수 없다. 각 LiDAR의 특성과 환경을 고려한 LiDAR 배치 연구는 다중 LiDAR 시스템을 활용하는데 있어서 필요할 것이다.
- 다중 LiDAR 오도메트리의 누적 오차를 해결할 수 있는 위치
 인식 알고리즘이 필요하다. 기존의 위치 인식 알고리즘은 단일

LiDAR에서 꾸준히 연구되었으며, 이 중에서도 회전 LiDAR를 주목표로 하여 연구 [37, 38]되어 왔다. Solid State LiDAR에 대한 연구 [39]도 있으나, 최근 들어 해당 연구가 시작되었기 때문에, 위치 인식에 대한 연구는 반드시 필요하다. 특히, 다중 LiDAR를 모두 종합하여, 최적의 위치 인식 해를 찾는 방식은 연구된 적이 없기 때문에 이에 대한 연구도 매우 중요할 것이다. 이와 같은 위치 인식은 다중 LiDAR 오도메트리에서 정확한 루프 클로저로 작용하여 누적된 오차를 줄여줄 수 있다는 점에서 의의가 있다.

- 각 LiDAR의 장점을 통합한 오도메트리 프레임워크의 연구가 가능하다. Velodyne LiDAR의 경우 주로 sparse하기 때문에 빠른 속도로 처리할 수 있다는 장점이 있고, Livox LiDAR의 경우 취득된 점군은 매순간은 저해상도이지만 정지한 상태에서 쌓이면 반복되지 않는 패턴 덕분에 고해상도의 지도를 얻을 수 있다. Aeva LiDAR의 경우 거리 정확도는 낮지만 주변 물체의 속도를 측정할 수 있다는 점에서 의의 [40]가 있으며, Ouster LiDAR의 경우 다른 LiDAR에서 확인할 수 없는 NIR, Reflectivity 등의 채널을 확인하여 사용할 수 있다. 해당 논문에서는 각 LiDAR의 강점이 일부만 활용되었지만, 지속적인 프레임워크 연구를 통해 해당 특성들을 활용하여 더욱 좋은 성능의 다중 LiDAR 오도메트리 프레임워크를 개발할 수 있을 것이다.
- 위의 연구들을 진행할 수 있는 다중 LiDAR 데이터셋이 부족하다. 대부분의 오픈 데이터셋은 오직 회전 LiDAR 또는 Solid State LiDAR 만을 포함하고 있으며, 함께 구성된 LiDAR 데이터셋의 경우에도 LiDAR 제조사의 종류가 많지 않거나, 매우 짧은 거리만의 데이터를 공개하고 있다. 따라서 City Dataset을 취득한 차량에 추가적인 LiDAR 센서를 설치하여 데이터를 취득하고, 위의 연구들에 활용할 것이다.

- [1] M. Helmberger, K. Morin, B. Berner, N. Kumar, G. Cioffi and D. Scaramuzza, "The Hilti SLAM Challenge Dataset," in IEEE Robotics and Automation Letters, vol. 7, no. 3, pp. 7518-7525, July 2022, doi: 10.1109/LRA.2022.3183759.
- [2] Hsu, Li-Ta, Kubo, Nobuaki, Wen, Weisong, Chen, Wu, Liu, Zhizhao, Suzuki, Taro, Meguro, Junichi, "UrbanNav:An Open-Sourced Multisensory Dataset for Benchmarking Positioning Algorithms Designed for Urban Areas," Proceedings of the 34th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2021), St. Louis, Missouri, September 2021, pp. 226-256. https://doi.org/10.33012/2021.17895
- [3] Nguyen T-M, Yuan S, Cao M, Lyu Y, Nguyen TH, Xie L. NTU VIRAL: A visualinertial-ranging-lidar dataset, from an aerial vehicle viewpoint. The International Journal of Robotics Research. 2022;41(3):270-280. doi:10.1177/02783649211052312
- [4] A. Reinke et al., "LOCUS 2.0: Robust and Computationally Efficient Lidar Odometry for Real-Time 3D Mapping," in IEEE Robotics and Automation Letters, vol. 7, no. 4, pp. 9043-9050, Oct. 2022, doi: 10.1109/LRA.2022.3181357.
- [5] T. -M. Nguyen, S. Yuan, M. Cao, L. Yang, T. H. Nguyen and L. Xie, "MILIOM: Tightly Coupled Multi-Input Lidar-Inertia Odometry and Mapping," in IEEE Robotics and Automation Letters, vol. 6, no. 3, pp. 5573-5580, July 2021, doi: 10.1109/LRA.2021.3080633.
- [6] Y. Wang, W. Song, Y. Lou, F. Huang, Z. Tu and S. Zhang, "Simultaneous Localization of Rail Vehicles and Mapping of Environment With Multiple LiDARs," in IEEE Robotics and Automation Letters, vol. 7, no. 3, pp. 8186-8193, July 2022, doi: 10.1109/LRA.2022.3187272.
- [7] J. Lv, K. Hu, J. Xu, Y. Liu, X. Ma and X. Zuo, "CLINS: Continuous-Time Trajectory Estimation for LiDAR-Inertial System," 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 2021, pp. 6657–6663, doi: 10.1109/IROS51168.2021.9636676.
- [8] D. Feng, L. Rosenbaum, and K. Dietmayer, "Towards safe autonomous driving: Capture uncertainty in the deep neural network for lidar 3d vehicle detection," in 2018 21st International Conference on Intelligent Transportation Systems (ITSC) (2018): 3266– 3273.
- [9] G. P. Meyer and N. Thakurdesai, "Learning an Uncertainty-Aware Object Detector for Autonomous Driving," 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 2020, pp. 10521-10527, doi: 10.1109/IROS45743.2020.9341623.
- [10] Y. Jung, M. Jeon, C. Kim, S. –W. Seo and S. –W. Kim, "Uncertainty–Aware Fast Curb Detection Using Convolutional Networks in Point Clouds," 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 2021, pp. 12882–12888, doi: 10.1109/ICRA48506.2021.9561358.
- J. Jiao, P. Yun, L. Tai and M. Liu, "MLOD: Awareness of Extrinsic Perturbation in Multi-LiDAR 3D Object Detection for Autonomous Driving," 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 2020, pp. 10556-10563, doi: 10.1109/IROS45743.2020.9341254.
- [12] J. Jiao, H. Ye, Y. Zhu and M. Liu, "Robust Odometry and Mapping for Multi-LiDAR Systems With Online Extrinsic Calibration," in IEEE Transactions on Robotics, vol. 38,

no. 1, pp. 351-371, Feb. 2022, doi: 10.1109/TRO.2021.3078287.

- B. Jiang and S. Shen, "A LiDAR-inertial Odometry with Principled Uncertainty Modeling," 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Kyoto, Japan, 2022, pp. 13292-13299, doi: 10.1109/IROS47612.2022.9981157.
- [14] W. Xu, Y. Cai, D. He, J. Lin and F. Zhang, "FAST-LIO2: Fast Direct LiDAR-Inertial Odometry," in IEEE Transactions on Robotics, vol. 38, no. 4, pp. 2053-2073, Aug. 2022, doi: 10.1109/TRO.2022.3141876.
- [15] T. Qin, P. Li and S. Shen, "VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator," in IEEE Transactions on Robotics, vol. 34, no. 4, pp. 1004-1020, Aug. 2018, doi: 10.1109/TRO.2018.2853729.
- [16] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time." in Robotics: Science and systems, vol. 2, no. 9. Berkeley, CA, 2014, p
- [17] T. Shan and B. Englot, "LeGO-LOAM: Lightweight and Ground-Optimized Lidar Odometry and Mapping on Variable Terrain," 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 2018, pp. 4758-4765, doi: 10.1109/IROS.2018.8594299.
- C. L. Gentil, T. Vidal-Calleja and S. Huang, "IN2LAMA: INertial Lidar Localisation And Mapping," 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 2019, pp. 6388-6394, doi: 10.1109/ICRA.2019.8794429.
- H. Ye, Y. Chen and M. Liu, "Tightly Coupled 3D Lidar Inertial Odometry and Mapping," 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 2019, pp. 3144-3150, doi: 10.1109/ICRA.2019.8793511.
- [20] T. -M. Nguyen, M. Cao, S. Yuan, Y. Lyu, T. H. Nguyen and L. Xie, "LIRO: Tightly Coupled Lidar-Inertia-Ranging Odometry," 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 2021, pp. 14484-14490, doi: 10.1109/ICRA48506.2021.9560954.
- [21] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti and D. Rus, "LIO-SAM: Tightlycoupled Lidar Inertial Odometry via Smoothing and Mapping," 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 2020, pp. 5135-5142, doi: 10.1109/IROS45743.2020.9341176.
- [22] M. Ramezani, K. Khosoussi, G. Catt, P. Moghadam, J. Williams, P. Borges, F. Pauling, and N. Kottege, "Wildcat: Online continuous-time 3d lidar-inertial slam," arXiv preprint arXiv:2205.12595, 2022.
- [23] W. Xu and F. Zhang, "FAST-LIO: A Fast, Robust LiDAR-Inertial Odometry Package by Tightly-Coupled Iterated Kalman Filter," in IEEE Robotics and Automation Letters, vol. 6, no. 2, pp. 3317-3324, April 2021, doi: 10.1109/LRA.2021.3064227.
- [24] C. Bai, T. Xiao, Y. Chen, H. Wang, F. Zhang and X. Gao, "Faster-LIO: Lightweight Tightly Coupled Lidar-Inertial Odometry Using Parallel Sparse Incremental Voxels," in IEEE Robotics and Automation Letters, vol. 7, no. 2, pp. 4861-4868, April 2022, doi: 10.1109/LRA.2022.3152830.
- [25] J. Liu, Y. Zhang, X. Zhao, and Z. He, "Fr-lio: Fast and robust lidar-inertial odometry by tightly-coupled iterated kalman smoother and robocentric voxels," arXiv preprint arXiv:2302.04031, 2023.
- [26] D. He, W. Xu, N. Chen, F. Kong, C. Yuan, and F. Zhang, "Point-lio: Robust highbandwidth light detection and ranging inertial odometry," Advanced Intelligent Systems, vol. 5, no. 7,pp. 2200459, 2023.
- [27] Y. Zhang et al., "RI-LIO: Reflectivity Image Assisted Tightly-Coupled LiDAR-Inertial

Odometry," in IEEE Robotics and Automation Letters, vol. 8, no. 3, pp. 1802-1809, March 2023, doi: 10.1109/LRA.2023.3243528.

- [28] H. Li, B. Tian, H. Shen and J. Lu, "An Intensity-Augmented LiDAR-Inertial SLAM for Solid-State LiDARs in Degenerated Environments," in IEEE Transactions on Instrumentation and Measurement, vol. 71, pp. 1–10, 2022, Art no. 8503610, doi: 10.1109/TIM.2022.3190060.
- [29] M. Palieri et al., "LOCUS: A Multi-Sensor Lidar-Centric Solution for High-Precision Odometry and 3D Mapping in Real-Time," in IEEE Robotics and Automation Letters, vol. 6, no. 2, pp. 421-428, April 2021, doi: 10.1109/LRA.2020.3044864.
- [30] T. -M. Nguyen, D. Duberg, P. Jensfelt, S. Yuan and L. Xie, "SLICT: Multi-Input Multi-Scale Surfel-Based Lidar-Inertial Continuous-Time Odometry and Mapping," in IEEE Robotics and Automation Letters, vol. 8, no. 4, pp. 2102-2109, April 2023, doi: 10.1109/LRA.2023.3246390.
- [31] Z. Wang, L. Zhang, Y. Shen and Y. Zhou, "D-LIOM: Tightly-coupled Direct LiDAR-Inertial Odometry and Mapping," in IEEE Transactions on Multimedia, doi: 10.1109/TMM.2022.3168423.
- [32] J. Lin, X. Liu and F. Zhang, "A decentralized framework for simultaneous calibration, localization and mapping with multiple LiDARs," 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 2020, pp. 4870-4877, doi: 10.1109/IROS45743.2020.9340790.
- [33] T. D. Barfoot and P. T. Furgale, "Associating Uncertainty With Three-Dimensional Poses for Use in Estimation Problems," in IEEE Transactions on Robotics, vol. 30, no. 3, pp. 679-693, June 2014, doi: 10.1109/TRO.2014.2298059.
- [34] M. Jung, S. Jung and A. Kim, "Asynchronous Multiple LiDAR-Inertial Odometry Using Point-Wise Inter-LiDAR Uncertainty Propagation," in IEEE Robotics and Automation Letters, vol. 8, no. 7, pp. 4211-4218, July 2023, doi: 10.1109/LRA.2023.3281264.
- [35] M. Grupp, "EVO: Python package for the evaluation of odometry and SLAM," 2017.
 [Online]. Available: https://github.com/MichaelGrupp/evo
- [36] H. Hu, Z. Liu, S. Chitlangia, A. Agnihotri and D. Zhao, "Investigating the Impact of Multi-LiDAR Placement on Object Detection for Autonomous Driving," in 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 2022 pp. 2540-2549. doi: 10.1109/CVPR52688.2022.00258
- [37] G. Kim, S. Choi and A. Kim, "Scan Context++: Structural Place Recognition Robust to Rotation and Lateral Variations in Urban Environments," in IEEE Transactions on Robotics, vol. 38, no. 3, pp. 1856–1874, June 2022, doi: 10.1109/TRO.2021.3116424.
- [38] T. Shan, B. Englot, F. Duarte, C. Ratti and D. Rus, "Robust Place Recognition using an Imaging Lidar," 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 2021, pp. 5469-5475, doi: 10.1109/ICRA48506.2021.9562105.
- [39] C. Yuan, J. Lin, Z. Zou, X. Hong and F. Zhang, "STD: Stable Triangle Descriptor for 3D place recognition," 2023 IEEE International Conference on Robotics and Automation (ICRA), London, United Kingdom, 2023, pp. 1897–1903, doi: 10.1109/ICRA48891.2023.10160413.
- [40] Y. Wu et al., "Picking up Speed: Continuous-Time Lidar-Only Odometry Using Doppler Velocity Measurements," in IEEE Robotics and Automation Letters, vol. 8, no. 1, pp. 264-271, Jan. 2023, doi: 10.1109/LRA.2022.3226068.

Abstract

Asynchronous Multiple LiDAR-Inertial Odometry using Pointwise Inter-LiDAR Uncertainty Propagation

Minwoo Jung

Department of Mechanical Engineering The Graduate School Seoul National University

Over the past few years, LiDAR has emerged as an essential sensor for location estimation and mapping. However, a single LiDAR has limitations in field of view (FOV) and object occlusion. Multiple LiDAR systems, offering higher accuracy and expanded FOV, have become popular but pose integration challenges due to temporal and spatial discrepancies. This paper presents a new approach to address these challenges by using the continuous-time trajectory with B-Spline interpolation. This method derives point-wise uncertainty, helping to alleviate strict time synchronization and overcome FOV differences. The proposed method is validated on open-source and proprietary datasets, showing compatibility with various LiDAR manufacturers and scanning patterns. The software is also opensourced on GitHub to promote further research.

Keywords : Multiple LiDAR , SLAM, Uncertainty, Asynchronous Student Number : 2021-24223