



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

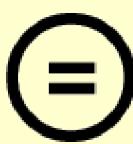
다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원 저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리와 책임은 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)



석사학위논문

3차원 객체 검출을 위한
포인트 클라우드 시퀀스의 시공간 표현 학습

Learning Spatiotemporal Representation of Point
Cloud Sequences for 3D Object Detection

이준형

한양대학교 대학원

2023년 2월

석사학위논문

3차원 객체 검출을 위한
포인트 클라우드 시퀀스의 시공간 표현 학습

Learning Spatiotemporal Representation of
Point Cloud Sequences for 3D Object Detection

지도교수 최준원

이 논문을 공학 석사학위논문으로 제출합니다.

2023년 2월

한양대학교 대학원

미래모빌리티학과

이준형

이 논문을 이준형의 석사학위 논문으로 인준함

2023년 2월

심사위원장 : 최정욱 (인)

심사위원 : 최준원 

심사위원 : 오윤선 

한양대학교 대학원

차례

차례.....	i
국문 요지.....	ii
제1장 서론.....	1
1.1 연구의 필요성.....	1
1.2 연구 목표.....	4
1.3 주요 연구 내용.....	4
제2장 관련 연구.....	7
2.1 포인트 클라우드를 사용한 딥러닝 기반 3차원 객체 검출기.....	7
2.2 포인트 클라우드 시퀀스에 대한 다중 스윕 전처리 기법.....	7
2.3 포인트 클라우드 시퀀스에 대한 종래 시공간 표현 알고리즘.....	9
제3장 Short-term Aware Grid Feature Encoder.....	10
3.1 알고리즘 연구 배경.....	10
3.2 Temporal-Channel Attention Network.....	11
3.3 Temporal bin-based Augmentation.....	13
제4장 Long-term BEV Feature Refinement.....	16
4.1 알고리즘 연구 배경.....	16
4.2 Motion-guided Deformable Alignment Network.....	20
4.3 Feature Aggregation by Alignment.....	24
제5장 검증 실험.....	25
5.1 nuScenes 데이터셋.....	25
5.2 포인트 클라우드 시퀀스 기반 알고리즘 실험 구성.....	25
5.3 SA-GFE 알고리즘 검증 실험.....	27
5.4 Long-term BEV feature refinement 알고리즘 검증 실험.....	33
5.5 LSR-3D 모델 검증 실험.....	37
제6장 결론.....	54
참고 문헌.....	55
Abstract.....	58

국문요지

물체 표면에 반사된 레이저 펄스를 바탕으로 포인트 클라우드를 생성하는 라이다 센서는 정확한 3 차원 공간 정보를 제공하기 때문에 자율주행과 로보틱스를 비롯한 여러 모빌리티 산업에서 인지(Perception) 기술 관련 핵심 센서 역할을 한다. 최근 딥러닝 기술이 빠르게 고도화됨에 따라 포인트 클라우드에서 특징 표현(Feature representation)을 도출하고 이를 3 차원 객체 검출(3D Object detection)에 활용하는 기술이 활발히 연구되고 있다. 한편, 시퀀스 형태의 데이터에 포함된 시공간(Spatiotemporal) 정보를 활용하는 알고리즘은 행동 인식, 객체 추적을 비롯한 여러 영상 인식 분야에서 그 효용성이 증명되어왔다. 라이다 센서 또한 연속된 스캐닝을 통해 실시간으로 포인트 클라우드 시퀀스를 만들어낸다. 하지만 종래 기술들은 단일 스캔 결과 기반의 3 차원 객체 검출 알고리즘을 중점적으로 다뤄왔으며, 시퀀스 데이터를 활용하더라도 단순 병합을 통해 얻은 좀 더 높은 밀도의 포인트 클라우드를 사용하는 것에 그치고 있다.

따라서 본 연구에서는 포인트 클라우드 시퀀스에 포함된 시공간 정보를 활용함으로써 라이다 센서 기반 객체 검출 성능 고도화에 한계 요인으로 분석되어 왔던 다음 두 가지 문제점을 개선하고자 했다. 첫째, 라이다 센서 해상도에 따른 포인트 데이터 희소성 문제를 개선하고자 했다. 둘째, 센서와 물체의 위치 관계에 따른 부분적 획득 문제를 개선하고자 했다. 이를 위해 본 연구에서는 포인트 클라우드 시퀀스를 단기 시퀀스와 장기 시퀀스로 구분된 계층적 구조로 바라보고, 계층적 관점에 기반해 시공간 특징 표현을 학습하고 활용할 수 있는 각각의 관점에 대한 딥러닝 알고리즘을 제안했다. 또한 포인트 클라우드 시퀀스 기반 3 차원 객체 검출 파이프라인을 설계하여 각각의 알고리즘을 종래 기술에 적용할 수 있도록 했다. 대표적인 자율주행 데이터셋에 해당하는 nuScenes 데이터셋을 사용해 제안된 알고리즘들에 대한 지도학습(Supervised learning)과 추론(Inference) 실험을 진행했으며, 실험 결과를 바탕으로 정량적 및 정성적 분석을 수행했다. 본 논문을 통해 제안된 검출 모델은 nuScenes 검증용 데이터셋에 대해 Baseline 모델인 PointPillars 대비 mAP 와 NDS 성능에서 각각 5.54%와 3.07%씩 성능 향상을 보임으로써 포인트 클라우드 시퀀스로부터 시공간 특징 표현을 활용하는 방식이 갖는 효용성을 증명했다. 특히 계층적 관점에 기반한 시공간 표현 학습을 통해 차량 관련 클래스에 비해 상대적으로 Bird's-eye-view 에서의 종횡비가 작고 취득되는 포인트 수가 적은 오토바이, 자전거 그리고 보행자 클래스에서 높은 검출 성능 향상을 보였다. 뿐만 아니라 정성적 분석을 통해 False positive 검출이 감소하고 객체간 중첩이 발생하는 경우에 대해서도 강건한 검출 성능을 보이는 것을 확인했다.

제1장 서 론

1.1 연구의 필요성

로보틱스 및 자율주행 분야에서 자율 이동체(Ego-vehicle)의 판단 및 제어를 위해 정확한 인지 시스템을 구성하는 것이 중요하다. 이때 라이다 센서는 카메라 센서에 비해 조도 변화, 기상 악화와 같은 각종 외란(Disturbance)에 강건하며, 레이더 센서에 비해 높은 밀도의 포인트 클라우드 형성이 가능하므로 인지 기술 고도화에 있어 핵심 센서에 해당한다. 특히 3차원 공간 상 정확한 거리 정보를 제공한다는 점에서 객체의 위치 및 크기를 예측하고(Localization), 동시에 어떤 종류인지 분류하는(Classification) 3차원 객체 검출(3D Object detection) 기술에 유용하다. 그럼 1은 포인트 클라우드 기반 3차원 객체 검출기의 입출력 관계를 보여준다. 최근 딥러닝 기술의 빠른 발달과 함께 포인트 클라우드를 활용한 딥러닝 기반 3차원 객체 검출 연구가 활발히 진행되고 있으며 본 연구 또한 관련 기술에 대한 내용을 다룬다.

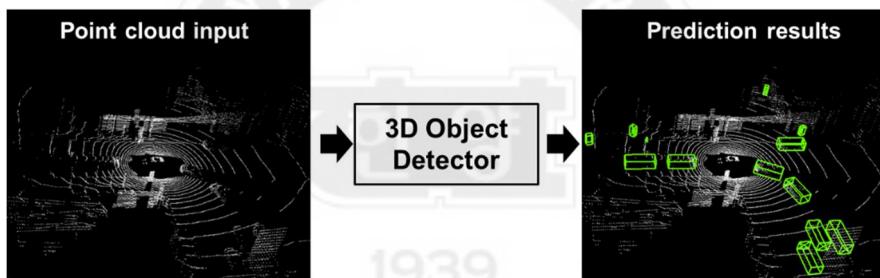


그림 1. 포인트 클라우드 기반 3차원 객체 검출기의 입출력 관계

객체 표면으로부터 반사된 레이저 펄스를 바탕으로 포인트 클라우드를 생성하는 라이다 센서 특성상 포인트 클라우드 기반 객체 검출 기술은 일반적으로 다음 두 가지 한계점이 존재한다. 첫째, 자율 이동체와 주변 객체들 간 위치 관계에 따라 객체 외형의 일부에 대한 포인트 취득만 가능하므로 정확한 검출이 어려우며(Partial-view problem), 객체들이 밀집되어 있어 중첩(Occlusion)이 발생할수록 어려움이 커진다. 둘째, 라이다 센서 해상도에 따른 포인트 클라우드 밀도 편차가 크게 발생하는데 해상도가 낮아질수록 성능 저하가 급격하게 발생한다. 하지만 고해상도 라이다 센서는 가격이 높아 상용화 관점에서 활용에 어려움이 있다.

실제 주행 환경에서 라이다 센서는 연속된 포인트 클라우드 데이터를 생성하므로 일정 시간 동안 취득된 스캔 결과, 즉 포인트 클라우드 시퀀스 데이터 활용이 가능하다. 최근 시퀀스 데이터를 객체 검출에 활용하기 위해 다중 스윕(Multi-sweep) 처리 방법이 제안되었다 [1]. 해당 방법은 연속된 포인트 클라우드 스캔 데이터들을 하나의 포인트 클라우드로 병합하여 검출기 입력으로 사용하는 포인트 클라우드 전처리 방식이다. 이때 해당 시구간의 가장 마지막 시점에 관한 라이다 좌표계를 기준으로

시퀀스 데이터가 병합되면 단일 스캔 데이터를 사용하는 경우보다 높은 밀도의 포인트 클라우드를 입력으로 활용할 수 있다.

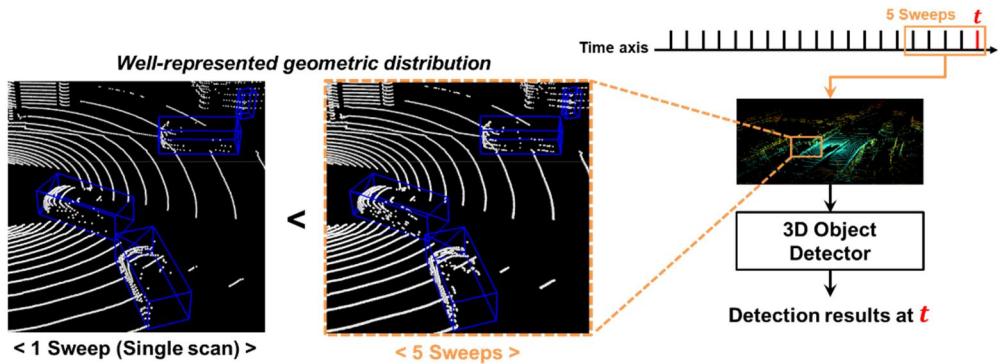


그림 2. 시점 t 에서의 검출 수행을 위한 다중 스윕 방식 적용 유무

그림 2는 검출기 입력으로 단일 스캔 결과만 사용하는 경우와 다중 스윕 방식을 통해 연속된 5개의 시퀀스 데이터를 병합하여 사용하는 경우를 시각적으로 비교해서 보여준다. 다중 스윕 방식을 적용하면 다음 두 가지 장점을 통해 객체 검출 성능을 높일 수 있다. 첫째, 연속된 스캔 결과를 병합하므로 포인트 밀도를 높일 수 있어 라이다 센서의 해상도에 따른 한계점 개선이 가능하다. 둘째, 병합된 포인트 클라우드 입력은 해당 시구간 동안 발생한 자율 이동체 및 주변 객체들의 움직임으로 인해 단일 스캔의 경우보다 시간에 따른 포인트 분포 정보를 많이 포함하고 있으므로 객체 검출을 위한 특징 표현 도출에 유리하다. 따라서 해당 장점은 부분적 취득 문제를 개선할 수 있다. 하지만 다중 스윕 방식을 적용하여 시퀀스 데이터를 병합함으로써 입력 포인트 클라우드 밀도를 높이고 포인트 분포의 공간적 정보량을 높이는 방식은 한계점이 있다.

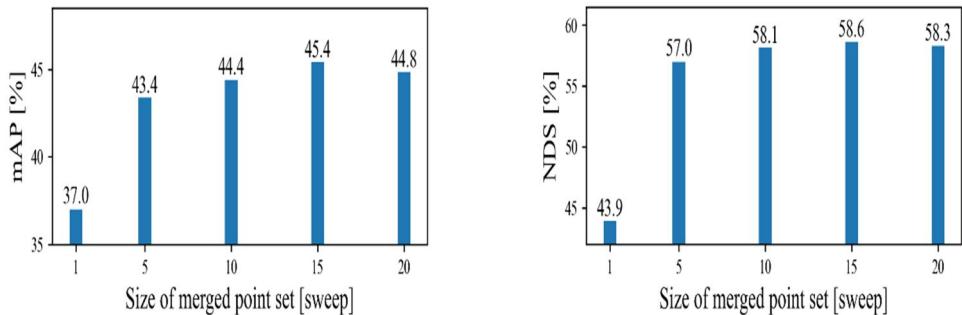


그림 3. nuScenes 데이터셋 [1]에 대한 PointPillars [2]의 3차원 객체 검출 성능

그림 3은 nuScenes 데이터셋 [1]에서 제공하는 라이다 센서의 포인트 클라우드 데이터를 이용해 병합에 사용된 스캔 데이터 개수, 즉 시퀀스 길이에 따른 3차원 객체

체 검출 성능 변화를 보여준다. 검출 성능 비교를 위해 mAP¹와 NDS² 두 가지 지표를 사용했으며 대표적인 딥러닝 기반 객체 검출기 PointPillars [2]를 사용해 선행 연구를 진행했다. nuScenes 데이터셋 [1]은 32 채널 해상도의 라이다 센서를 이용해 20Hz로 스캐닝한 결과를 제공하며 Sweep은 다중 스윕 방식을 통해 병합된 시퀀스 길이를 가리킨다. 1sweep은 시퀀스 병합을 수행하지 않은 단일 스캔 결과를 사용해 검출을 수행한 경우를 가리킨다. 15sweep, 즉 0.75초 분량의 시퀀스 데이터를 사용한 경우까지는 시퀀스 길이가 늘어날수록 검출 성능이 높아졌다. 하지만 20sweep 분량을 병합하여 입력으로 사용한 경우, 더 많은 포인트 클라우드 데이터를 사용했음에도 불구하고 성능 저하가 발생했다.

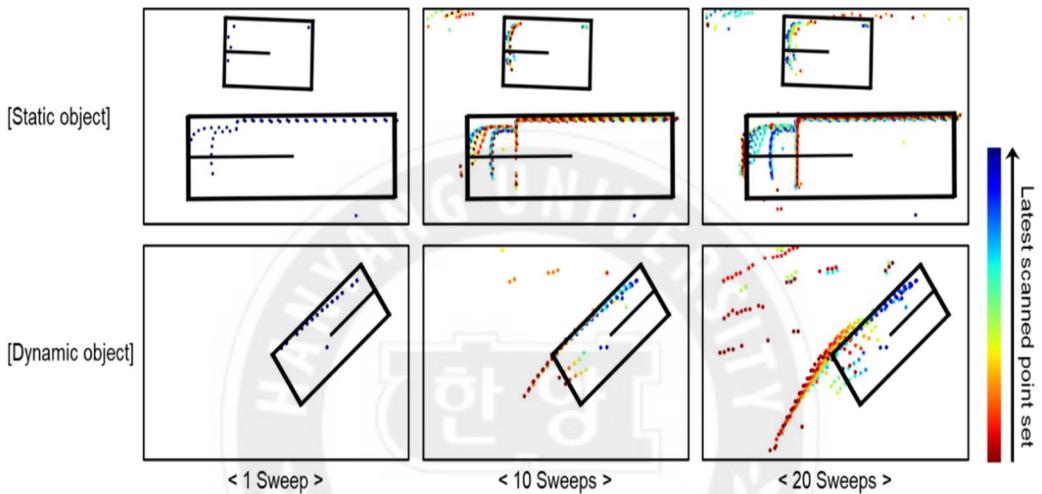


그림 4. Sweep 분량에 따른 객체 유형별 포인트 클라우드 분포

성능 저하 요인은 시퀀스 길이에 따른 병합된 포인트 클라우드의 공간적 분포를 Bird's eye-view (BEV)에서 시각화한 그림 4를 통해 확인할 수 있다. 다중 스윕 방식은 병합 과정에서 자율 이동체 움직임 보상을 수행하는데, 이때 병합에 사용된 시퀀스 길이가 늘어날수록 포인트 분포 번짐 정도 (Point distribution smearing)가 심해지는 것을 그림 4에서 알 수 있다. 특히 정적 객체 (Static object)에 비해 동적 객체 (Dynamic object)에서 해당 현상은 두드러진다. 객체의 움직임으로 인한 포인트 번짐 현상은 주변 객체에 대한 검출기의 Localization 정확도를 낮추는 요인으로 분석 가능하다. 따라서 그림 3에 제시된 선행 연구 결과와 그림 4의 정성적 분석을 통해 다중 스윕 방식은 긴 시간 범위의 시퀀스 데이터를 활용하는데 한계점이 있음을 알 수 있다. 또한 병합을 통해 높은 밀도의 포인트 클라우드 입력을 사용하는 것은 검출기 파이프라인에서 전처리 기술에 해당하므로 딥러닝 기반 검출기 연구 관점에서

¹ mAP: mean average precision

² NDS: nuScenes detection score

볼 때, 이는 시퀀스 데이터에 포함된 시공간 정보를 충분히 활용하지 못한 것으로 바라볼 수 있다.

비디오, 즉 여러 장의 이미지를 가지고 해당 시퀀스 데이터에 담긴 시공간 영역 정보를 활용하는 것은 행동 인식(Action recognition), 2차원 비디오 객체 검출(2D VOD³) 등 여러 컴퓨터 비전 분야에서 그 효용성이 증명되어왔다 [3,4,5]. 또한 딥러닝 기술을 적용해 시공간 영역 정보를 도출하는 알고리즘들이 활발히 연구되고 있다. 반면 3차원 객체 검출을 위해 포인트 클라우드 시퀀스 데이터로부터 시공간 영역 정보를 도출하고 이를 활용하는 연구는 상대적으로 미비한 편에 속한다. 따라서 포인트 클라우드 시퀀스로부터 시공간 표현(Spatiotemporal representation)을 학습하고 객체 검출 성능 향상을 위해 이를 활용하는 딥러닝 알고리즘 연구가 필요하다.

1.2 연구 목표

본 논문에서 다루는 연구는 포인트 클라우드 기반 3차원 객체 검출기의 성능 고도화를 목표로 진행되었다. 이를 위해 연구 과정은 크게 2단계로 구분된다. 먼저, 포인트 클라우드 시퀀스 데이터로부터 시공간 표현을 학습하기 위한 딥러닝 알고리즘을 설계하였다. 기존 3차원 객체 검출기에 설계한 알고리즘들을 적용한 뒤, 자율주행 데이터셋을 이용해 지도 학습 및 검증에 관한 실험을 진행하였으며 정량적 및 정성적 분석을 통해 제안된 시공간 표현 학습의 효용성 확인했다.

1.3 주요 연구 내용

포인트 클라우드 시퀀스에 포함된 시공간 특징 표현(Spatiotemporal feature representation)을 학습하고 이를 활용하여 객체 검출 성능을 향상시키고자 본 연구는 포인트 클라우드 시퀀스 입력을 계층적(Hierarchical) 관점으로 바라본다.

계층적 관점이란 주어진 시퀀스를 단기 시퀀스와 장기 시퀀스, 두 가지 관점으로 구분한 관점이다. 단기 시퀀스란 포인트 클라우드 시퀀스 데이터가 다중 스윕 전처리를 통해 병합된 단일 프레임⁴에 해당하는 시구간을 가리키며, 장기 시퀀스란 단기 시퀀스가 여러 개 모인 다중 프레임(Multi frame)에 해당하는 시구간을 가리킨다. 그림 5는 12개의 시퀀스 데이터에 대해 4 sweep 단위의 단기 시퀀스가 3개 모여 장기 시퀀스를 형성한 것으로 바라보는 계층적 관점의 예시를 보여준다.

³ VOD: Video object detection

⁴ 본 연구에서는 다중 스윕 방식을 통해 병합된 포인트 클라우드 시퀀스 입력을 단일 프레임으로 정의.

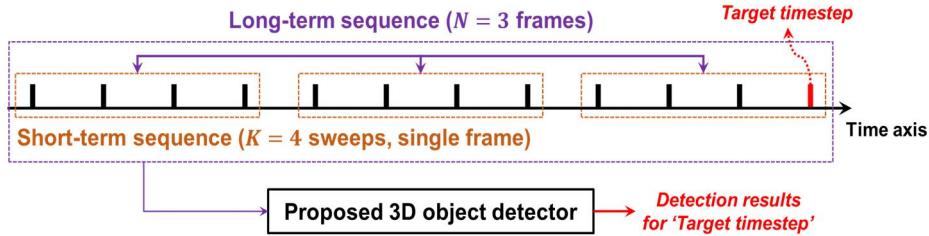


그림 5. 포인트 클라우드 시퀀스에 대한 계층적 관점 예시

본 연구에서는 입력으로 K sweep 단위의 단일 프레임이 N 개 모인 포인트 클라우드 시퀀스 데이터, 즉 총 $(N \times K)$ 개의 시퀀스 데이터를 사용해 Target timestep⁵에 대한 검출 결과를 예측하는 3차원 객체 검출기, LSR-3D⁶를 제안한다. LSR-3D는 시퀀스 데이터에 대한 두 가지 관점으로부터 시공간 특징 표현을 학습하기 위한 딥러닝 알고리즘을 포함하고 있다. 그림 6은 각 관점에 대응하는 시공간 영역을 설명하고 있다. 단기 시퀀스 알고리즘을 통해 격자 형태의 3차원 공간 영역과 단일 프레임에 해당하는 시간 영역으로부터 시공간 표현을 학습하고, 장기 시퀀스 알고리즘을 통해 BEV 공간 영역과 다중 프레임에 걸친 시간 영역으로부터 시공간 표현을 학습한다.

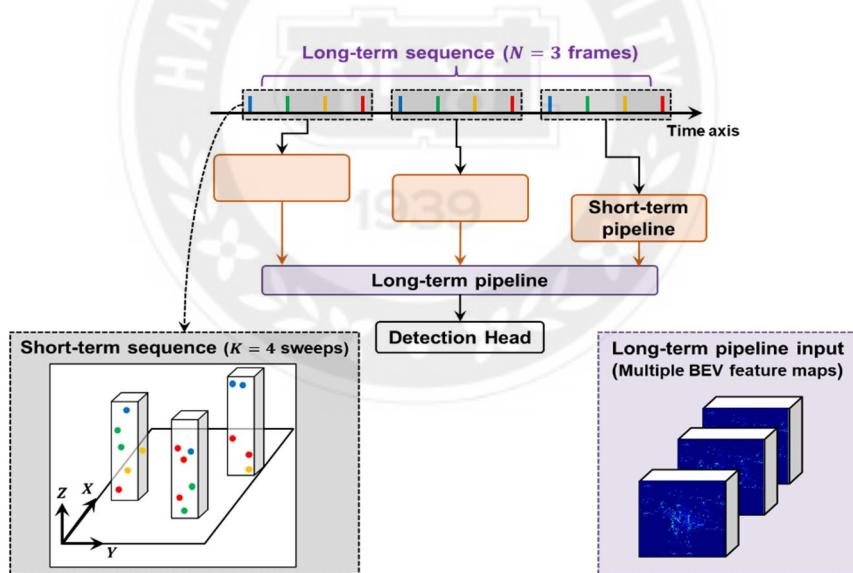


그림 6. 계층적 관점에 기반한 포인트 클라우드 데이터의 공간 영역과 시간 영역

제3장에서는 단기 시퀀스에 대한 시공간 표현 알고리즘을 설명하고 제4장에서는

⁵ 객체 검출 목표 시점. 검출기가 입력으로 사용한 포인트 클라우드 시퀀스 중에서 가장 마지막에 해당하는 시점을 나머지 시점들과 구분하기 위해 Target timestep이라 정의.

⁶ LSR-3D: Learning spatiotemporal representation of point cloud sequences for 3D object detection

장기 시퀀스 관점에서 설계한 시공간 표현 알고리즘을 설명한다. 제5장에서는 제3장과 제4장에서 제안된 알고리즘들 각각에 대한 검증 실험과 알고리즘이 모두 통합된 LSR-3D 모델의 객체 검출 성능에 대한 검증 실험 결과를 제시한다.



제2장 관련 연구

2.1 포인트 클라우드를 사용한 딥러닝 기반 3차원 객체 검출기

딥러닝 기반 3차원 객체 검출기는 검출에 필요한 특징 추출 과정에 앞서, 3차원 포인트 클라우드 입력에 대해 사전 정의된 크기의 격자 단위로 복셀화(Voxelization)를 수행한다. 3차원 직육면체 형태의 격자를 Voxel이라 하며, 이때 사전 정의된 Voxel의 높이가 복셀화 영역에 대해 지면 수직방향 축의 길이와 동일한 경우 Pillar라고 지칭한다. 그림 7은 Pillar 단위의 복셀화 결과를 바탕으로 객체 검출 결과를 예측하는 PointPillars [2] 모델 파이프라인을 보여준다. 주어진 격자 구조에 대해 격자별 특징 벡터를 도출하여, 4D 또는 3D Tensor 형태의 특징 표현을 얻은 다음 (Grid feature encoding⁷), 이를 3D 및 2D Convolutional layer 기반 Backbone network를 통과시켜 고도화된 의미론적(Semantic) 특징을 얻는다. 해당 특징 표현은 BEV feature map, BEV representation 등의 표현으로 지칭되며 이는 Backbone network에서 지면의 수직방향 축에 관한 차원을 축소하는 Convolutional 연산을 수행하기 때문이다. 최종적으로 객체에 대한 Localization과 Classification을 수행하기 위해 BEV feature map을 Detection head에 전달해 객체별로 직육면체 형태의 검출 결과를 예측한다.

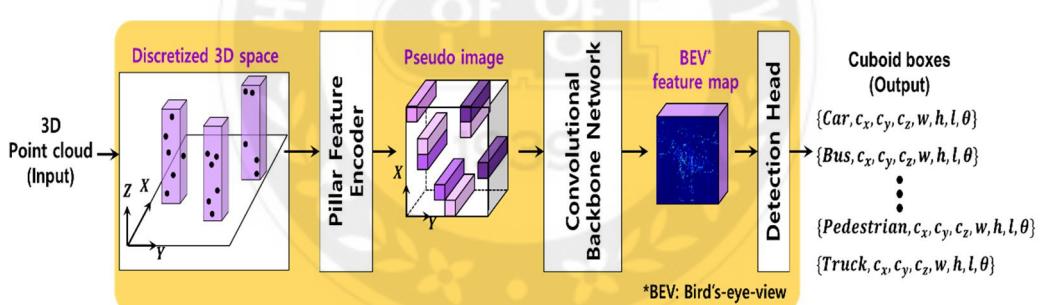


그림 7. PointPillars 모델 파이프라인 구조

2.2 포인트 클라우드 시퀀스에 대한 다중 스윕 전처리 기법

nuScenes [1]에서 제안된 다중 스윕 방식은 입력 포인트 클라우드 시퀀스의 가장 마지막 시점을 기준으로 자율 이동체 움직임 보상을 적용해 하나의 포인트 클라우드 형태로 병합하는 방식이다. 이는 검출기 입력을 구성하는 전처리 기법에 해당하므로 종래 관련 연구에서는 해당 기법을 기본 전처리 방식으로 활용해왔다. 연속된 N 개의 시퀀스 데이터에 대해 병합 기준 시점을 t 로 지정하고 기준 시점보다 과거 시점

⁷ PointPillars [2]의 경우, Pillar 형태의 격자를 사용하므로 Pillar feature encoding이라 함.

들을 각각 $t - 1, t - 2, \dots, t - N + 1$ 이라 하자. 이때 자율 이동체가 $t - n$ 시점 위치에서 t 시점 위치에 이르기까지 움직인 거리를 반영하는 회전 및 이동 행렬을 $t - n$ 시점에 취득한 포인트 클라우드 P_{t-n} 에 일괄 반영하여 t 시점(병합 기준 시점)의 포인트 클라우드에 추가하는 식으로 병합한다($n \in \{1, 2, \dots, N - 1\}$). 본 연구에서는 다중 스윕 방식을 사용하는 종래 기술들과 마찬가지로 병합 기준 시점에 해당하는 데이터를 Key sample이라고 지칭한다. 그럼 8은 입력 포인트 클라우드 시퀀스를 4개 단위로 병합하는 과정을 보여준다.

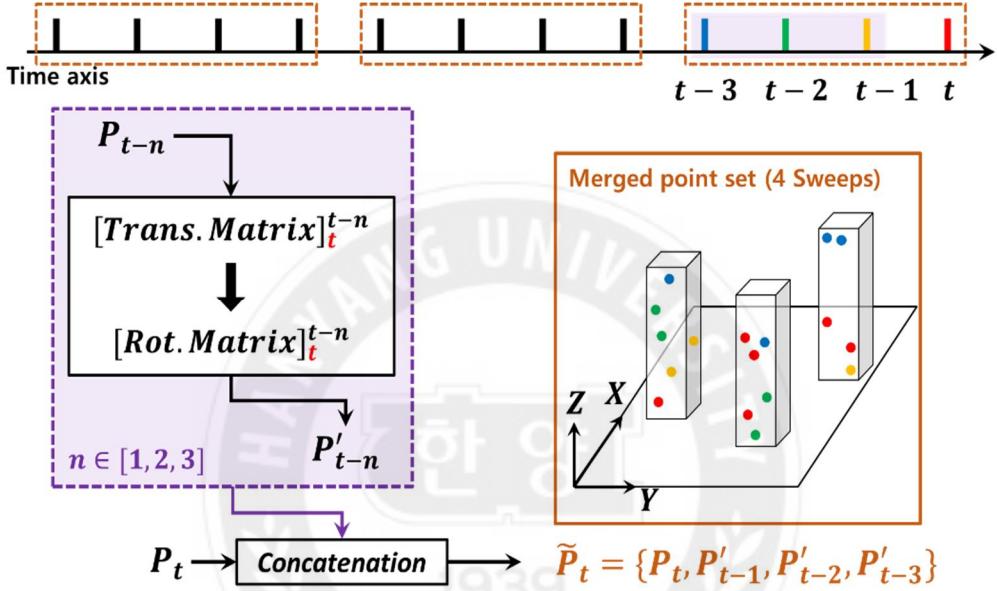


그림 8. 다중 스윕 전처리 기반 포인트 클라우드 입력 구성 예시

여러 시점의 포인트들이 함께 포함된 3차원 포인트 클라우드에 대해 각각의 포인트들에 대한 취득 시점을 구분하고자 Time offset 정보를 추가로 사용한다. 이때 Time offset은 기준 시점과 과거 시점의 차이로 계산된다. 그럼 8에서 살펴보면 병합 기준 시점 t 에 대해, $t - n$ 시점 포인트 클라우드에 적용된 Time offset은 n 이다. 즉, $t - n$ 시점 포인트 클라우드 P_{t-n} ⁸의 i 번째 포인트 $p_{t-n,i}$ 를 $[x_{t-n,i}, y_{t-n,i}, z_{t-n,i}, \gamma_{t-n,i}] \in \mathbb{R}^4$ 로 표현할 때, P'_{t-n} , 즉 기준 시점에 맞게 변환된 P_{t-n} 의 i 번째 포인트 $p'_{t-n,i}$ 는 $[x'_{t-n,i}, y'_{t-n,i}, z'_{t-n,i}, \gamma_{t-n,i}, n] \in \mathbb{R}^5$ 로 표현된다. 뿐만 아니라 병합 기준 시점 t 를 포함해 4개의 시점에 대한 시퀀스 데이터가 병합된 포인트 클라우드 \tilde{P}_t 는 모두 \tilde{C}_t 개의 포인트가 포함⁹된 집합이 된다.

⁸ $P_{t-n} = \{p_{t-n,i}\}_{i=1}^{C_{t-n}}$, C_{t-n} 은 P_{t-n} 에 포함된 총 포인트 수

⁹ $\tilde{C}_t = \sum_{n=0}^3 C_{t-n}$

2.3 포인트 클라우드 시퀀스에 대한 종래 시공간 표현 알고리즘

포인트 클라우드 시퀀스로부터 시공간 영역 특징 정보를 학습하고 이를 객체 검출에 활용하는 알고리즘은 최근 여러 연구들을 통해 그 효용성이 검증되었다.

VelocityNet [6]은 포인트 클라우드 시퀀스에 포함된 속도 정보로부터 움직임 특징 정보를 도출한 뒤 이를 활용한 Time-deformed convolution을 제안하였고 연속된 시퀀스를 다중 스윕 방식으로 병합하여 한 번에 Convolution을 수행하는 경우보다 우수한 성능을 보였다.

시퀀스 데이터로부터 특징 표현을 얻는데 주로 사용되는 LSTM [7]과 GRU [8] 구조에 포함된 연산 방식을 2차원 영역으로 확장한 ConvLSTM [9]과 ConvGRU [10]를 포인트 클라우드 시퀀스 모델링에 활용한 연구도 있다. LSTM_TOD [11]는 ConvLSTM [9]을 3차원 영역에 맞게 변형시킨 시공간 표현 알고리즘을 제안했다. 3DVID [12]는 Self-attention 기법을 ConvGRU [10]에 적용시킨 알고리즘을 통해 단일 스캔 데이터를 입력으로 사용한 경우보다 높은 검출 성능을 도출했다.

TCTR [13]은 특정 표현들간 관계 모델링에 장점을 보인 Transformer [14] 구조를 시간-채널 영역에 대한 관계 모델링에 적용하여 포인트 클라우드 시퀀스 데이터로부터 시공간 표현을 학습하는 알고리즘을 제안했다.

검출기 파이프라인에서 Detection head 단계를 거치기 전 특정 표현들을 가공하는데 초점을 둔 상기 방식들과는 달리 3D-MAN [15]은 Two-stage 구조의 알고리즘을 통해 시공간 정보를 활용했다. 각각의 단일 스캔 데이터에서 도출된 Object proposal들에 대한 특징 표현을 Memory bank에 저장해두고 Proposal들 간 특징 관계를 파악하여 검출 시점에 관한 Proposal들의 특징 표현을 정제함으로써 (Refinement) 성능을 높였다.

제3장 Short-term Aware Grid Feature Encoder

3.1 알고리즘 연구 배경

단기 시퀀스 관점 기반 시공간 특징 표현 학습의 필요성

다중 스윕 전처리 기법이 적용된 포인트 클라우드 입력에 대해 각각의 포인트들은 초기 특징 벡터(Initial feature vector)에 3차원 좌표와 라이다 반사율 이외에도 Time offset 값이 함께 포함되어 있다.¹⁰ 격자 별 특징 벡터를 도출하는 딥러닝 파이프라인에서 Time offset 정보를 함께 기본 정보로 사용하는 건 그 자체로 해당 파이프라인에서 시공간 정보를 활용하는 것으로 볼 수 있다. 하지만 그림 3과 그림 4에 제시된 선행 연구 결과를 통해 해당 정보만으로는 보다 긴 시간 범위의 포인트 클라우드 시퀀스 데이터를 활용하는데 한계가 있음을 확인했다. 따라서 제3장에서는 주어진 격자 공간 영역과 단기 시퀀스 시간 영역에 포함된 시공간 특징 표현을 학습하기 위한 딥러닝 알고리즘을 제안한다.

Short-term aware grid feature encoder(SA-GFE)는 종래 다중 스윕 방식을 통해 병합되는 시간 범위, 즉 단일 프레임 입력으로부터 시공간 표현을 도출하는 알고리즘이다. SA-GFE 알고리즘은 종래 격자 기반 3차원 객체 검출기들의 전체 파이프라인 중 3차원 포인트 클라우드 입력을 Voxel 또는 Pillar 형태의 격자 단위로 구분한 뒤, 각 격자별로 특징 표현을 도출하는 파이프라인에 적용할 수 있다.

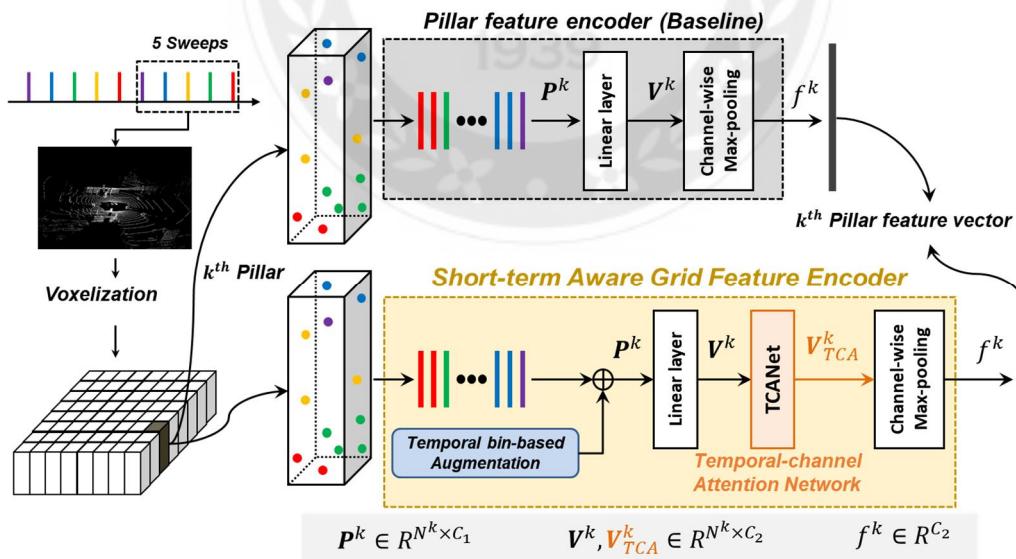


그림 9. 종래 Pillar feature encoder와 SA-GFE 알고리즘 비교

¹⁰ 2.2절 내용 참고

3.2 Temporal–Channel Attention Network

Temporal–channel attention network(TCANet)는 종래 Grid feature encoder¹¹ 구조에 Plug-and-play 형태로 추가하여 시공간 특징 표현을 학습할 수 있는 딥러닝 알고리즘이다. 그림 9는 PointPillars [2]의 Pillar feature encoder(PFE)에 제안된 TCANet을 적용한 경우를 비교해서 보여주고 있으며 다중 스윕 전처리를 통해 5개의 스캔 결과를 병합하여 단일 프레임 입력으로 사용하는 경우를 다룬다.

N^k 는 k 번째 Pillar에 포함된 포인트 개수를 가리키며 C_1 과 C_2 는 각각 포인트 별 특정 벡터의 채널 수(차원)을 가리킨다. 예를 들어 각각의 포인트들에 대한 초기 특정 벡터를 포인트의 3차원 좌표 $\in \mathbb{R}^3$, 라이다 반사율 $\in \mathbb{R}^1$, Time offset $\in \mathbb{R}^1$ 으로 구성했다면, C_1 은 5에 해당한다. 또한 Linear layer의 출력 채널 수에 따라 C_2 가 결정된다. SA-GFE 알고리즘에서 TCANet의 역할은 Linear layer를 통해 1차적으로 Encoding된 포인트별 특정 벡터 행렬 \mathbf{V}^k 에 대해 Channel-wise maxpooling 연산을 수행하기 앞서 Temporal–channel attention을 통해 \mathbf{V}_{TCA}^k 를 도출하는 것이다.

T sweep 단위로 병합된 포인트 클라우드 입력에 대해 복셀화를 수행한 결과, k 번째 Pillar에 대한 포인트 입력 집합 $\mathbf{P}^k \in \mathbb{R}^{N^k \times C_1} (= \{\mathbf{p}_t^k \in \mathbb{R}^{N_t^k \times C_1} | 1 \leq t \leq T\})$ 를 얻는다. 이때, \mathbf{p}_t^k 는 해당 Pillar에 포함된 시점 t 에 관한 포인트 입력 집합을 가리키며, N_t^k 는 시점 t 에 관한 포인트 수를 가리킨다 ($N^k = \sum_{t=1}^T N_t^k$). 따라서 그림 10에 묘사된 k 번째 Pillar에 대한 Linear layer 출력 \mathbf{V}^k 는 식 (1)로 표현 가능하다.

$$\mathbf{V}^k \in \mathbb{R}^{N^k \times C_2} = [\mathbf{v}_1^k, \mathbf{v}_2^k, \dots, \mathbf{v}_t^k, \dots, \mathbf{v}_T^k] \quad (1)$$

$\mathbf{v}_t^k \in \mathbb{R}^{N_t^k \times C_2}$ 는 시점 t 에 대한 포인트 특징 행렬을 가리킨다.

TCANet은 주어진 \mathbf{V}^k 에 대해 두 가지 방향의 Attention 연산을 수행한다. 첫 번째, Temporal–wise attention(TWA)을 통해 병합된 길이 T 의 시퀀스 중 어느 시점에 스캔 된 포인트들에 대한 특징 벡터에 가중치를 둬야 하는지 결정한다. 두 번째, Channel–wise attention(CWA)을 통해 포인트 특징 벡터의 채널 중 어느 채널이 군집을 Encoding하는데 중요한지 결정한다. 이를 위해, Attention을 주고자 하는 각 방향에 대한 특징 응답(Feature response)을 도출한다. 길이 T 의 시퀀스에 대한 Temporal–wise response $u^k \in \mathbb{R}^{T \times 1}$ 는 각각의 $\mathbf{v}_t^k (t \in [1, T])$ 에 대해 Point–wise max pooling과 Channel–wise maxpooling을 함께 적용하여 얻는다. 또한 Channel–wise maxpooling을 \mathbf{V}^k 에 적용하여 Channel–wise response $g^k \in \mathbb{R}^{1 \times C_2}$ 를 얻는다. 그럼 n 의 Temporal–channel wise response extraction 부분에서 상기 과정을 설명하고 있다. 도출된 각각의 특징 응답에 대해 SENet [16]에서 제안된 Squeeze–and–excitation 연산을 적용해 가중치 벡터 $b^k \in \mathbb{R}^{T \times 1}$ 와 $h^k \in \mathbb{R}^{1 \times C_2}$ 를 얻는 과정을 식 (2)와 (3)에서 보이고 있다.

$$b^k = W_{TWA}^2 \delta(W_{TWA}^1 u^k) \quad (2)$$

¹¹ Pillar feature encoder 또는 Voxel feature encoder

$$h^k = (W_{CWA}^2 \delta(W_{CWA}^1(g^k)^T))^T \quad (3)$$

$W_{TWA}^1 \in \mathbb{R}^{\gamma \times T}$, $W_{TWA}^2 \in \mathbb{R}^{T \times \gamma}$, $W_{CWA}^1 \in \mathbb{R}^{\gamma \times C_2}$ 그리고 $W_{CWA}^2 \in \mathbb{R}^{C_2 \times \gamma}$ 는 모두 학습 가능한 행렬 파라미터에 해당하며, γ 는 $\frac{C_2}{2}$ 로 설정했다. 또한 $\delta(\cdot)$ 는 ReLU 함수를 가리킨다.

두 가지 가중치 벡터에 대해 Element-wise multiplication와 Sigmoid 함수를 적용해 0에서 1사이의 값을 갖는 가중치 행렬 $M^k \in \mathbb{R}^{T \times C_2} (= \{m_t^k \in \mathbb{R}^{1 \times C_2} | 1 \leq t \leq T\})$ 를 얻을 수 있으며, m_t^k 는 시점 t 에 대한 포인트 특징 행렬에 적용하기 위한 가중치 벡터를 가리킨다. 식 (4)를 통해 시간-채널 축으로 가중치가 반영된 특징 표현 \tilde{V}_{TCA}^k 를 얻는다.

$$\tilde{V}_{TCA}^k = [\mathbf{v}_1^k \otimes m_1^k, \mathbf{v}_2^k \otimes m_2^k, \dots, \mathbf{v}_t^k \otimes m_t^k, \dots, \mathbf{v}_T^k \otimes m_T^k] \quad (4)$$

TCANet을 통해 도출된 시공간 특징 표현 \tilde{V}_{TCA}^k 는 \mathbf{V}^k 와 Point-wise concatenation을 통해 결합 후 Linear layer를 통해 k 번째 Pillar에 대한 최종적인 포인트 특징 행렬 \mathbf{V}_{TCA}^k 를 얻는다. 그럼 10은 \mathbf{V}^k 로부터 특징 응답을 얻는 것부터 가중치 행렬을 통해 시공간 특징 표현 \mathbf{V}_{TCA}^k 을 얻기까지 일련의 과정을 보여준다.

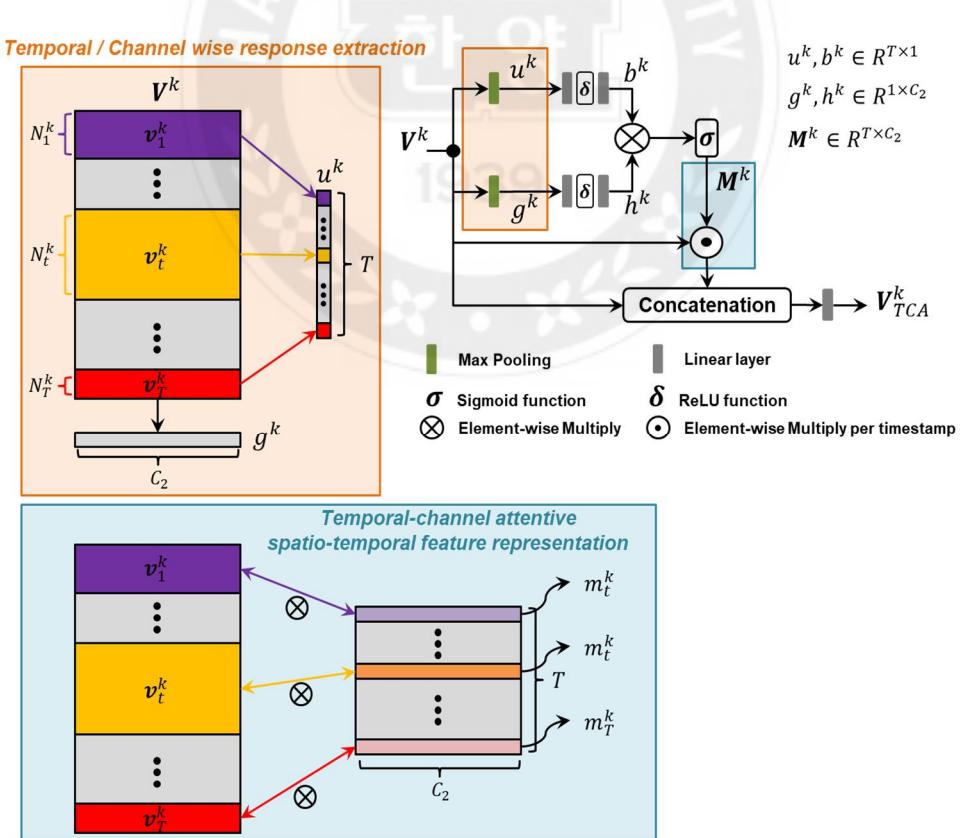


그림 10. TCANet 구조와 연산 과정

그림 9에 제시되었듯이 k 번째 Pillar에 대한 대표 특징 벡터를 얻기 위해 시공간 특징 표현 \mathbf{V}_{TCA}^k 에 Channel-wise maxpooling을 적용해 f^k 를 얻는다. 각 격자 별로 도출된 특징 벡터가 모여 얻어진 4D 또는 3D Tensor 형태의 특징 표현은 종래 객체 검출기에서 사용하는 Convolutional backbone network와 Detection head를 통해 객체 검출 결과를 예측하는데 사용된다.

3.3 Temporal bin-based Augmentation

단기 시퀀스 범위의 단일 프레임 입력에 포함된 각 포인트들에 대한 초기 특징 벡터는 라이다 스캔 정보와 복셀화 결과를 바탕으로 구성된다. 라이다 스캔 정보에는 해당 포인트의 3차원 좌표, 라이다 반사율 그리고 Key sample 시점과 해당 포인트가 스캔 된 시점에 대한 Time offset 정보가 포함된다. 또한 복셀화를 수행하면 해당 포인트가 포함된 Voxel 또는 Pillar 내부 포인트들 사이의 기하적 관계 정보가 함께 활용될 수 있다. 이는 포인트 특징 벡터에 대해 딥러닝 기반 Encoding을 하는데 있어 앞서 언급한 기본적인 라이다 스캔 정보에 추가할 수 있는 정보 증대 (Augmentation)의 일부다. 종래 기술들이 사용하는 관계 정보는 격자 구조 형태를 기반으로 계산 가능한 두 가지 좌표를 기준으로 얻는다. 첫 번째 종류의 좌표는 대상 포인트가 포함된 격자의 절대 중심 좌표에 해당하고, 두 번째 종류의 좌표는 해당 격자 내 포인트들이 갖는 산술 평균 좌표에 해당한다. 해당 좌표들을 기준으로 격자 내 포인트들의 상대적 L1 distance 값을 계산하여 이를 라이다 스캔 기본 정보 이외에도 각각의 포인트들에 대한 초기 포인트 특징 정보로 활용한다. 그림 11은 5sweep 단위로 구성된 포인트 클라우드 입력에 대해 복셀화를 수행한 결과 임의의 격자에 포함된 특정 포인트의 초기 특징 벡터를 구성하는데 있어 기하적 관계 정보를 계산하는 과정을 설명하고 있다. Key sample 시점을 t_1 라 표기하고 앞선 스캔 시점에 대해 순서대로 t_2 부터 t_5 로 표기했다. k 번째 격자에 대해 시점 t_5 에 취득된 i 번째 포인트의 초기 특징 벡터 $p_{t_5,i}^k \in \mathbb{R}^{11}$ 는 해당 포인트의 라이다 스캔 정보와 기하적 정보가 함께 구성됨을 알 수 있다. 해당 포인트의 Time offset t'_5 는 t_1 과 t_5 의 차이로 계산되며 격자의 중심 좌표와 격자 내 포인트들이 갖는 산술 평균 좌표는 각각 c 와 a 로 표기하였다.

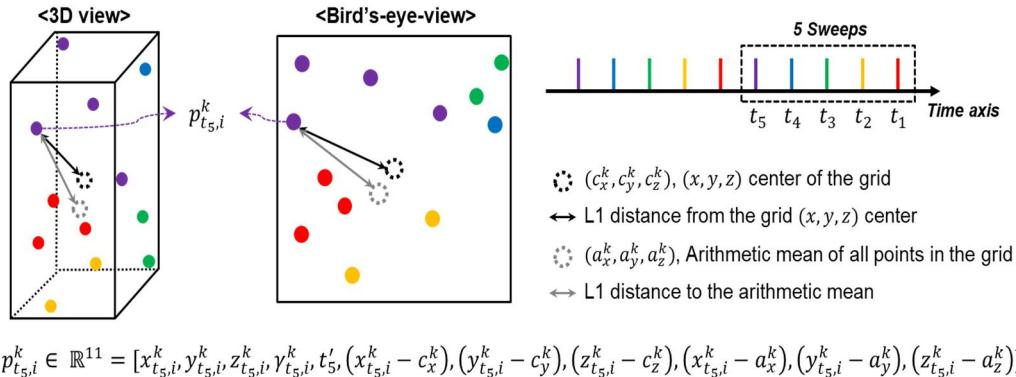


그림 11. 종래 기술의 포인트별 초기 특징 벡터 구성 방식

하지만 종래 기술에서 적용하는 산술 평균 위치와의 거리 계산은 격자 내 병합된 포인트들을 모두 반영하므로 공간 정보를 도출하는데 있어 시간 정보가 전혀 활용되지 않는 한계점이 있다. 따라서 3.3절에서는 Temporal bin 개념을 바탕으로 산술 평균 좌표를 추가로 활용함으로써 시간 정보도 함께 반영된 포인트별 초기 특징 벡터에 관한 Augmentation 기법을 제안한다.

Temporal bin이란 병합된 포인트 클라우드 시퀀스를 일정한 시구간으로 균등하게 구분하는 것을 가리킨다. T sweep 단위로 포인트 클라우드 시퀀스를 병합해 검출기 입력으로 사용하는 경우 주어진 시퀀스를 균등하게 분할하기 위해서 T 의 약수에 해당하는 값으로 Temporal bin 길이를 설정 가능하다. 본 연구는 nuScenes¹² 데이터셋을 대상으로 알고리즘을 개발했으므로 10sweep 단위의 입력을 사용하는 가정하에 Temporal bin 도입 및 적용에 대해 설명한다. 먼저 10의 약수는 2와 5가 있으므로 Bin의 크기를 2와 5로 설정한다. 이를 통해 10개의 포인트 클라우드 시퀀스는 각각 어느 Bin에 포함되는지 정해지며, 각각의 Bin에 포함된 포인트들을 대상으로 산술 평균 좌표를 구할 수 있다. Bin 크기가 2인 경우 모두 5개의 산술 평균 좌표가 얻어지며, 5인 경우 2개의 산술 평균 좌표가 얻어진다. 만약 Bin 크기를 Sweep 단위에 해당하는 10으로 설정한다면, 앞서 설명했던 종래 기술의 기하적 관계 정보에 포함된 산술 평균 좌표를 얻는 셈이다. 그림 12는 10개의 포인트 클라우드 시퀀스에 대해 주어진 Bin 크기를 바탕으로 어떻게 구획하고, 임의의 격자에 포함된 포인트들로부터 산술 평균 좌표를 구해 각 포인트에 대한 L1 distance 계산 과정을 묘사하고 있다.

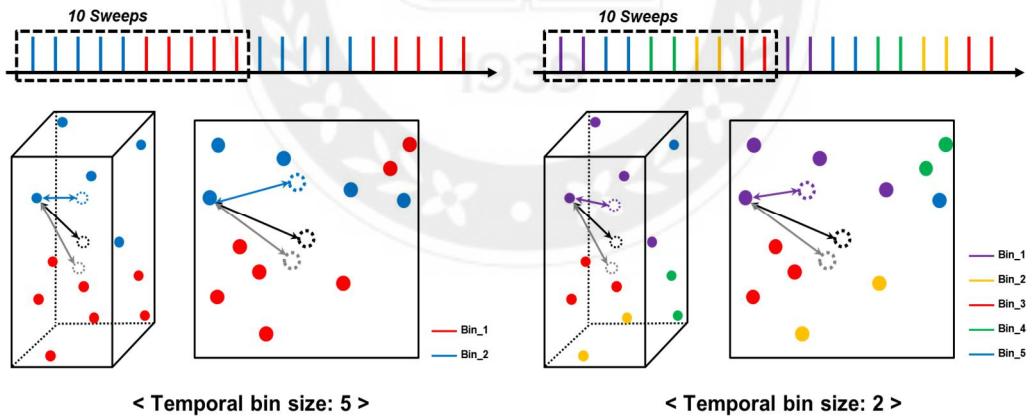


그림 12. Temporal bin 기반 산술 평균 좌표 계산 과정

k 번째 격자에 포함된 시점 t_n 의 i 번째 포인트 $p_{t_n,i}^k$ 에 대해 각각의 해당하는 Bin에

¹² Key sample 포함 9개의 스캔 결과를 병합한 포인트 클라우드 시퀀스를 라이다 기반 객체 검출기 입력으로 제공.

서 얻은 3차원 산술 평균 좌표를 $[a_{x,2}^k, a_{y,2}^k, a_{z,2}^k] \in \mathbb{R}^3$ 와 $[a_{x,5}^k, a_{y,5}^k, a_{z,5}^k] \in \mathbb{R}^3$ 라 할 때, L1 distance 값 $[(x_{t_n,i}^k - a_{x,2}^k), (y_{t_n,i}^k - a_{y,2}^k), (z_{t_n,i}^k - a_{z,2}^k)] \in \mathbb{R}^3$ 와 $[(x_{t_n,i}^k - a_{x,5}^k), (y_{t_n,i}^k - a_{y,5}^k), (z_{t_n,i}^k - a_{z,5}^k)] \in \mathbb{R}^3$ 가 종래 기술에서 사용하던 초기 특징 벡터에 추가되어 $p_{t_n,i}^k \in \mathbb{R}^{11}$ 대신 $p_{t_n,i}^k \in \mathbb{R}^{17}$ 를 사용할 수 있다. 따라서 Temporal bin을 활용하면 포인트 클라우드 시퀀스에 포함된 공간 정보를 다양한 시간 관점에서 도출할 수 있으며 종래 기술에서 활용하던 정보에 추가되어 Feature encoding 파이프라인¹³에 좀 더 높은 정보량을 갖는 포인트 별 초기 특징 벡터를 전달할 수 있다.



¹³ 격자 별 포인트 특징 벡터를 Encoding하는 부분. 제3장에서 알고리즘을 적용하고자 하는 파이프라인이며 그림 7의 Pillar feature encoder에 해당.

제4장 Long-term BEV Feature Refinement

4.1 알고리즘 연구 배경

장기 시퀀스 관점 기반 시공간 특징 표현

포인트 클라우드 기반 3차원 객체 검출기 파이프라인 전체를 놓고 볼 때, 제3장의 SA-GFE는 격자 내 포함된 단기 시퀀스 분량의 포인트 클라우드로부터 시공간 표현을 학습하고 이를 활용함으로써 종래 기술의 Feature encoding 알고리즘 성능을 높이고자 제안된 것이다. 즉 해당 알고리즘은 단일 프레임 입력에 대한 시공간 특징 표현 알고리즘에 해당한다. 종래 격자 기반 객체 검출기에 관한 연구들은 주로 BEV 공간에서 검출 결과를 예측하는 구조를 사용한다. 하지만 그림 13에 제시된 예시처럼 포인트 클라우드에 대한 BEV 공간의 특징 표현은 객체에 관한 특징 분포가 희소하며 이는 라이다 센서의 해상도가 낮아질수록 두드러진다. 따라서 검출 정확도 향상을 위해 BEV 공간의 특징 표현을 강화하는(BEV feature refinement) 알고리즘이 다양한 형태로 제안되어 왔다. [17,18,19]

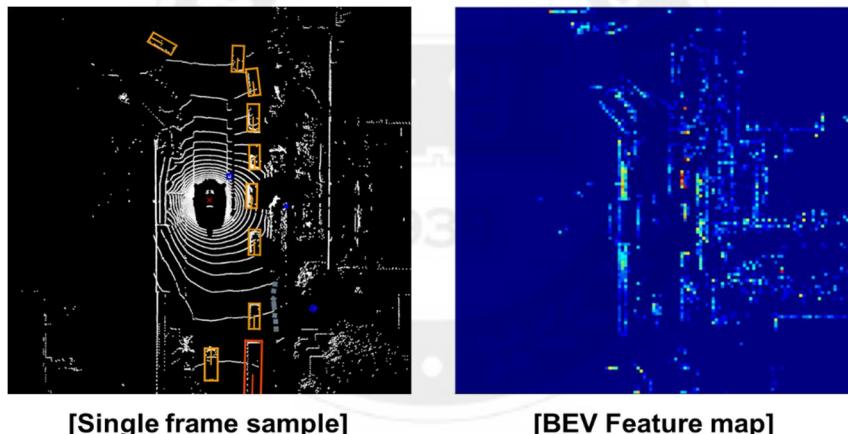


그림 13. 단일 프레임 입력에 대한 BEV 공간에서의 특징 표현

본 연구에서는 다중 프레임(Multi-frame) 입력, 즉 장기 시퀀스 관점에서 BEV 공간의 특징 표현을 강화하는 알고리즘을 제안한다. 구체적으로 제4장에서 제안하는 알고리즘은 다중 프레임 입력에 대한 단기 시퀀스 파이프라인의 출력¹⁴, 즉 연속된 시점에 대한 BEV feature map들로부터 시공간 표현을 학습하고 이를 활용한다. 본 연

¹⁴ 격자 기반 3차원 객체 검출기는 Grid feature encoding 결과, 격자 형태가 Voxel 또는 Pillar 형태인지에 따라 4D 또는 3D Tensor 형태의 특징 표현을 얻게 되며, 이를 Convolutional layer 기반 파이프라인에 통과시켜 BEV feature map을 도출한다. 2.1절 참고

구에서는 다중 프레임 입력을 구성하는 각 프레임의 역할을 구분하기 위해 Target timestep¹⁵이 포함된 단일 프레임 입력을 Target 프레임이라 정의하고 나머지 입력 프레임들을 Support 프레임이라 정의한다. 3개의 연속된 프레임을 사용하는 경우에 대한 예시를 그림 14에서 보여준다. 장기 시퀀스 관점 알고리즘의 목표는 Target 프레임과 Support 프레임에 관한 BEV feature map들을 가지고 시공간 특징 표현을 학습해 특징을 결합하는 것이며(Feature aggregation), 이는 결과적으로 객체 검출에 필요한 Target frame의 특징 표현¹⁶을 BEV 공간상에서 강화하는(Refinement) 과정으로 볼 수 있다.

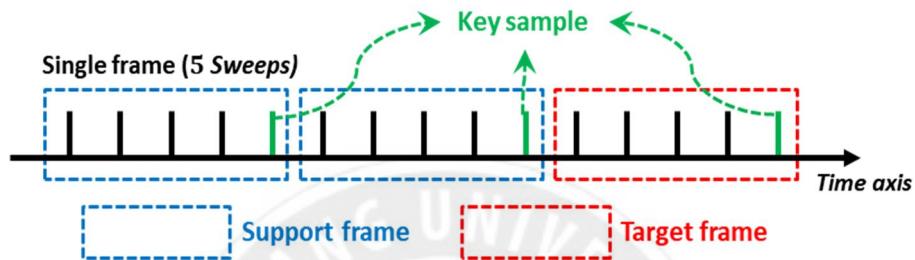


그림 14. 장기 시퀀스를 구성하는 각 프레임에 대한 구분

프레임간 BEV 공간 특징 변위

장기 시퀀스 관점에서 시공간 특징 표현을 학습하고 활용하는데 있어 본 연구에서 중요하게 바라본 문제는 연속된 시점의 BEV Feature map 들 사이에 존재하는 시간에 따른 프레임간 공간적 특징 변위(Inter-frame spatial feature displacement) 문제다. 다중 스윕 전처리 방식을 기반으로 단일 프레임 입력을 구성할 때, Key sample 기준으로 자율 이동체 움직임 보상을 적용한다. 다시 말해 해당 단기 시퀀스 길이 단위로 포인트 좌표들이 정렬된 채 격자 별 Feature encoding 과 Convolutional backbone network 를 통해 프레임별 BEV feature map 이 도출된다. 이는 인접한 프레임 사이에 자율 이동체를 중심으로 한 주변 객체들의 위치가 BEV 공간상에서 정렬되지 않았음을 의미¹⁷한다. 그림 15 는 인접한 두 프레임에 관한 BEV feature map 과 BEV 에서 바라본 객체의 실제 위치 관계를 비교해서 BEV 공간상 특징 값들의 위치 변위를 정성적으로 보여준다. 인접한 두 프레임의 Key sample 시점 사이에 발생한 자율 이동체의 움직임으로 인해 정적 객체(Static object)에 관한 Activated feature 의 BEV 공간상 변위가 존재하며, 이러한 BEV 공간상의 Feature 위치 변위는 개별적인 움직임을 보이는 동적 객체(Dynamic object)에서 더 크게 발생하는 것을 알 수 있다. 따라서 장기 시퀀스

¹⁵ 1.4 절 Target timestep 정의 참고

¹⁶ BEV feature map을 Detection head로 전달하여 객체 검출 수행, 그림 7 참고

¹⁷ 각각의 단일 프레임에 대한 Ego-vehicle 위치를 중심으로 3차원 공간 Voxelization 수행

관점에서 여러 개의 BEV feature map 을 결합할 때, Target 프레임을 기준으로 특정 변위를 조정하는 것에 대한 고려가 필요하다.

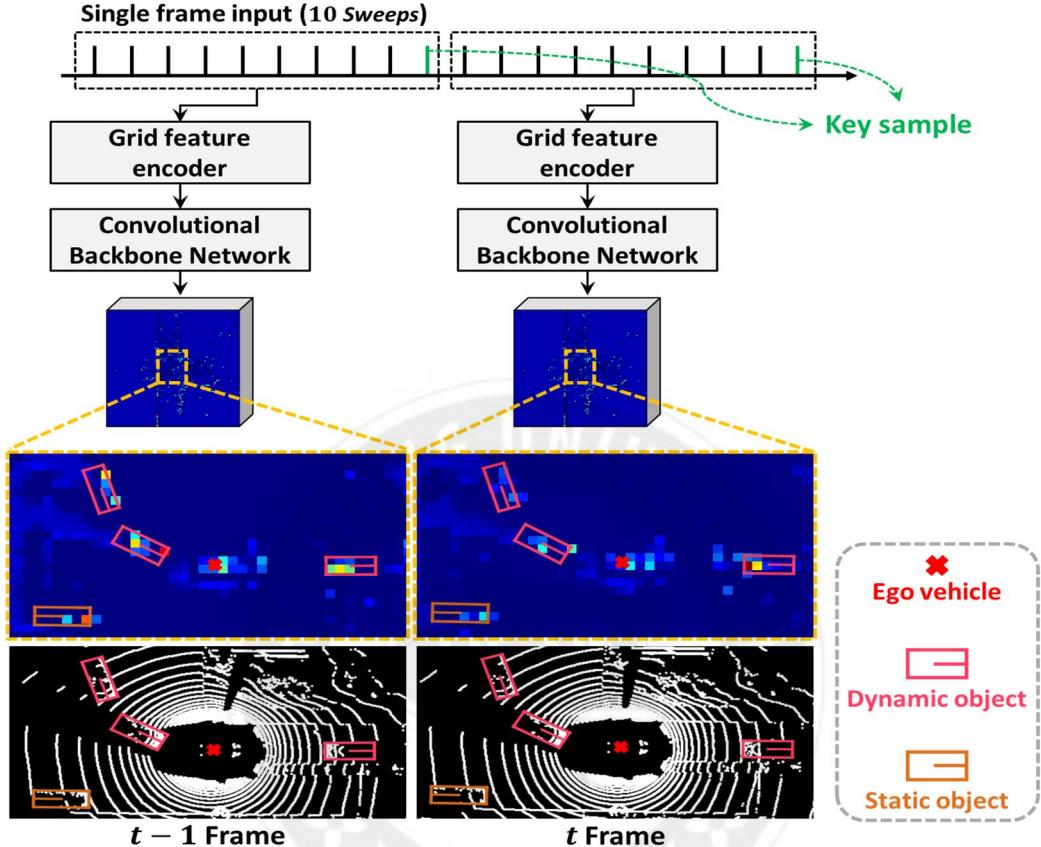


그림 15. 인접한 프레임간 BEV 공간에서의 Feature displacement 예시

인접한 프레임간 특징 변위를 조정하는 것, 즉 특징 정렬(Feature alignment)의 필요성을 정량적으로 확인하기 위해 간단한 선행 연구를 진행했다. 다중 프레임 입력에 대해 PointPillars [2]의 Backbone network로부터 얻은 BEV feature map 들을 가지고 간단한 특징 결합을 수행한 뒤 Detection head 에 전달함으로써, 프레임간 BEV 공간에서의 위치 변위 문제가 객체 검출 성능에 주는 영향을 확인했다. nuScenes 데이터셋을 사용했으므로 단일 프레임은 10 sweep 단위로 구성했다. 선행 연구에서 사용한 파이프라인은 다음과 같다. n 번째 프레임에 포함된 포인트 개수를 I_n 라 할 때, 연속된 N 프레임의 포인트 클라우드 입력 $\{\mathbf{P}_n \in \mathbb{R}^{I_n \times C_1}\}_{n=t-N+1}^t$ 에 대한 BEV feature map 집합 $\{\mathbf{F}_n \in \mathbb{R}^{C_2 \times H \times W}\}_{n=t-N+1}^t$ 은 식 (5)를 통해 얻어진다. t 는 주어진 다중 프레임 입력의 Target frame 인덱스를 가리킨다.

$$\mathbf{F}_n = G(\mathbf{P}_n) \quad , n \in \{(t - N + 1), \dots, (t - 1), t\} \quad (5)$$

함수 $G(\cdot)$ 는 Pillar feature encoding 과 Backbone network 까지 일련의 단기 시퀀스에 대한 파이프라인을 가리키며, C_1 과 C_2 는 각각 초기 포인트 특징 벡터의 채널 수와 BEV feature map 의 채널 수를 가리킨다. 주어진 N 개의 BEV feature map 에 대해 식 (6)을 통해 특징 결합을 수행하면 시공간적으로 결합된 특징 $\mathbf{F}'_t \in \mathbb{R}^{C_2 \times H \times W}$ 를 얻을 수 있고 이를 Detection head 에 전달해 객체 검출 결과를 예측한다.

$$\mathbf{F}'_t = \text{Conv}_{3 \times 3}([\mathbf{F}_t, \mathbf{F}_{t-1}, \dots, \mathbf{F}_{t-N+1}]) \quad (6)$$

$\text{Conv}_{3 \times 3}(\cdot)$ 은 3×3 커널 기반 Convolution 연산에 Batch Normalization [20]과 ReLU 함수가 이어지는 Convolutional layer 를 가리키며, $[\cdot, \cdot]$ 는 Channel-wise concatenation 을 가리킨다. 그럼 16 은 $N = 3$ 인 경우에 대한 특징 결합 과정을 묘사하고 있다. 선행 연구에 관한 검출 모델 학습과 평가는 입력 프레임 수 N 값을 변화시켜가며 진행했으며¹⁸ 검출 성능 결과는 표 1 에 제시되어 있다.

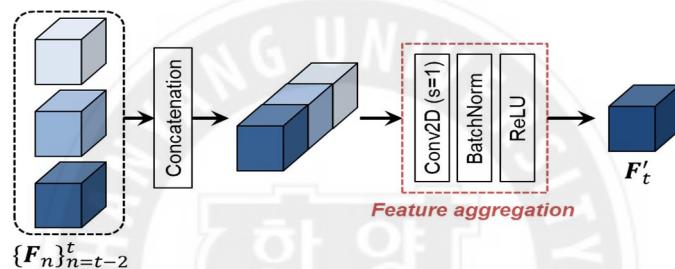


그림 16. Concatenation 기반 특징 결합 방식

	Baseline	<i>Multi frame-based detector</i>			
Sequence length [frame]	1	2	3	4	
Feature aggregation		✓	✓	✓	
mAP [%]	44.4	43.9(-0.5)	42.57(-1.83)	37.67(-6.73)	
NDS [%]	58.15	57.96(-0.19)	57.06(-1.09)	54.53(-3.62)	

표 1. 입력 프레임 수에 따른 검출 성능 결과

PointPillars [2]를 단일 프레임 입력 기반 Baseline 검출기로 사용했으며, 입력 프레임 길이 N 의 경우는 Target 프레임과 $(N - 1)$ 의 과거 시점 Support 프레임을 사용한 경우를 가리킨다. 장기 시퀀스상에서 사용하는 Support 프레임 수가

¹⁸ 선행 연구에 관한 모델 학습 및 평가 방식은 5.2절 지도학습 및 평가 부분을 따른다.

늘어날수록 mAP 와 NDS 성능 모두 저하되는 실험 결과를 통해 특징 정렬을 수행하기 위한 시공간 표현 학습의 필요성을 확인할 수 있다.

4.2 Motion-guided Deformable Alignment Network

일반적인 Convolution 연산의 경우, 주어진 Kernel 에 대해 사전 정의된 위치로부터 입력 Feature map 의 특징 값을 추출한 뒤¹⁹ 해당 값을 가지고 연산을 수행한다. 한편, BEV 공간상에서 자율 이동체 주변 객체들은 제각기 다른 방향과 속도의 움직임 패턴(Motion pattern)을 보인다. 따라서 Local receptive field 를 갖는 기존 Convolution 연산은 다양한 Motion pattern 으로 비롯된 인접한 프레임간 공간적 위치 변위를 조정하기에 한계점이 존재한다. 표 1에서 확인했듯이 두 프레임간 시간 간격이 커질수록 연관된 특징 값의 위치 변위가 커지므로 한계점은 더욱 두드러진다.

제 4 장에서는 Target 프레임의 BEV feature map \mathbf{F}_t 을 중심으로 나머지 Support 프레임의 BEV feature map $\{\mathbf{F}_n\}_{n=t-N+1}^{t-1}$ 을 정렬하기 위해 Deformable convolution (DCN) [21] 기반 특징 정렬 알고리즘, Motion-guided deformable alignment network(MDANet)을 제안한다. DCN [21]은 Sampling offset 과 Modulation scalar 로 구성된 Deformable mask 를 고려한 Convolution 연산을 수행함으로써 다양한 범위(Range)와 강도(Intensity)의 Receptive field 가 반영 가능한 Convolution 연산이다. 이때, Sampling offset 은 Input feature map 에 대한 특징 값의 추출 위치를 결정하는 요소에 해당하며, Modulation scalar 는 Convolution 연산 수행에 앞서 추출된 특징 값에 대한 조정 값에 해당한다. MDANet 은 인접한 프레임간 BEV 공간에서 도출된 Motion pattern 정보를 바탕으로 Deformable mask 를 예측하고(Deformable mask estimator) 이를 Convolution 연산에 반영하여 정렬된 BEV feature map 집합 $\{\mathbf{F}_n^{align}\}_{n=t-N+1}^{t-1}$ 을 출력한다. MDANet 의 Deformable mask estimator 는 적절한 Deformable mask 예측에 필요한 Motion pattern 정보를 도출하기 위해 다중 스케일(Multi-scale) BEV feature map 을 활용하도록 설계되었다. 의미론적(Semantic) 특징을 도출하는데 있어 주어진 공간에 대한 여러 개의 크기에 걸친 Feature map 을 사용하는 방식은 기존 객체 검출 연구 들에서 이미 그 효용성이 증명되었다. [22,23] 또한 제안하는 방식은 그림 17 에 묘사된 Backbone network 파이프라인의 중간 출력 값에 해당하는 Feature map 들을 사용하기 때문에 이를 얻기 위한 별도의 파이프라인을 두지 않고 그대로 사용 가능하다. Backbone network 에서 생성하는 BEV feature map 의 Scale 종류 수를 S 라 할 때, $\{\mathbf{F}_n^1, \mathbf{F}_n^2, \dots, \mathbf{F}_n^S\}_{n=t-N+1}^t$ 은 N 프레임에 대한 Multi-scale feature 집합을 가리킨다. 그림 17 은 $S=3$ 인 PointPillars [2] 의 Backbone network 를 묘사하고 있다.

¹⁹ 3x3 Kernel과 Dilation값이 1로 정의된 2D Convolution연산의 경우, 위치 p_0 에 대한 9개의 Feature sampling 위치는 $(p_0 + p_i)$ 에 해당하며, p_i 는 집합 $\{(-1,-1), (-1,0), \dots, (0,1), (1,1)\}$ 의 원소를 가리킨다.

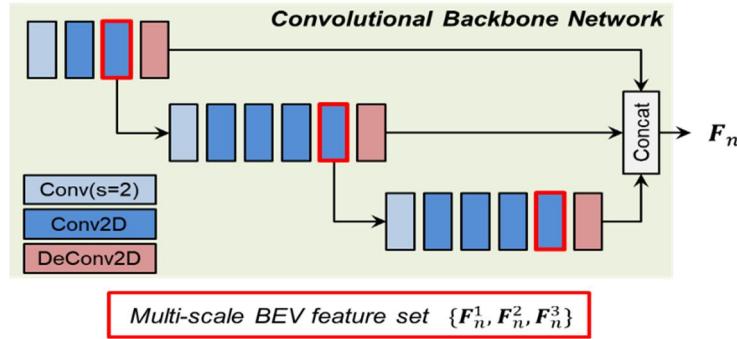


그림 17. PointPillars 모델의 Convolutional backbone network

$S = 3$ 인 경우, $t - n$ 번째 프레임의 BEV feature map \mathbf{F}_{t-n} 에 적용하기 위한 Deformable mask 예측 과정은 다음과 같다. 먼저 식 (7), (8)을 통해 각 Scale에서의 BEV 공간에 대한 Motion feature \mathbf{M}_{t-n}^s 를 얻는다.

$$\hat{\mathbf{M}}_{t-n}^s = \text{Conv}_{3 \times 3}([\mathbf{F}_t^s, \mathbf{F}_{t-n}^s]) \quad , s \in \{1, 2, 3\} \quad (7)$$

$$\mathbf{M}_{t-n}^s = \hat{\mathbf{M}}_{t-n}^s + \text{NLblock}(\hat{\mathbf{M}}_{t-n}^s) \quad , s \in \{1, 2, 3\} \quad (8)$$

NLblock(\cdot)은 Non-local block [24] 연산을 가리키며 넓은 범위에 걸친 Motion pattern 정보를 얻고자 활용했다. 서로 다른 Scale에서 도출된 Motion feature 집합 $\{\mathbf{M}_{t-n}^s\}_{s=1}^3$ 에 대해 그림 18에서 묘사하는 바와 같이 Deconvolution, 즉 Upsampling 연산을 적용해 결합을 수행함으로써 $\mathbf{F}_{t-n} \in \mathbb{R}^{C \times H \times W}$ 와 동일한 scale을 갖는 $\mathbf{M}_{t-n} \in \mathbb{R}^{C \times H \times W}$ 를 얻는다.

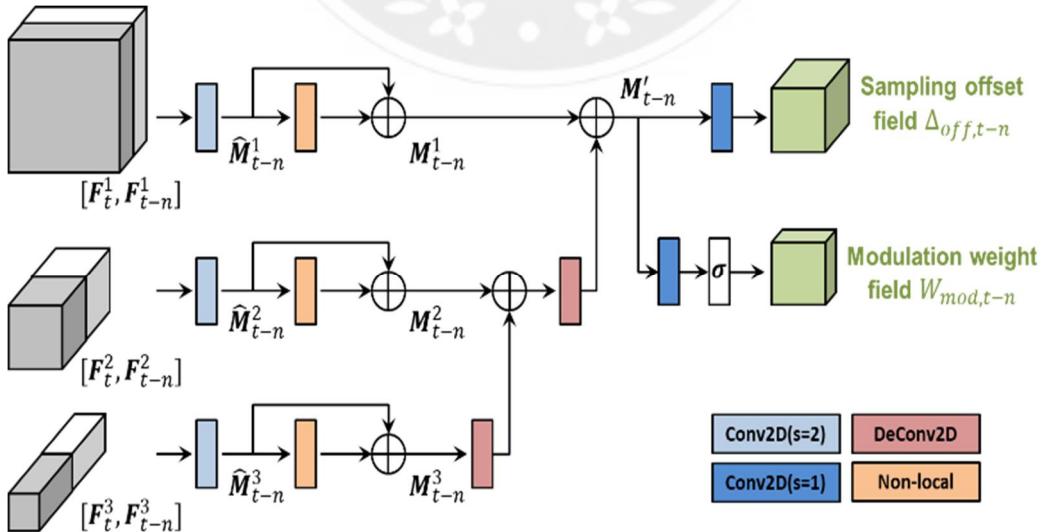


그림 18. Multi-scale BEV feature map 기반 Deformable mask 도출 과정

인접한 두 프레임에 대한 여러 Scale 의 Motion pattern 이 담긴 \mathbf{M}'_{t-n} 를 가지고 식 (9), (10)을 통해 Sampling offset field $\Delta_{t-n} \in \mathbb{R}^{2B \times H \times W}$ 와 Modulation weight field $W_{mod,t-n} \in \mathbb{R}^{B \times H \times W}$ 를 도출한다.

$$\Delta_{t-n} = \text{Conv}'_{3 \times 3}(\mathbf{M}'_{t-n}) \quad (9)$$

$$W_{mod,t-n} = \sigma(\text{Conv}'_{3 \times 3}(\mathbf{M}'_{t-n})) \quad (10)$$

$\text{Conv}'_{3 \times 3}(\cdot)$ 은 Stride 와 Padding 값이 모두 1 인 3×3 Convolution 연산을 가리킨다. $\sigma(\cdot)$ 는 Sigmoid 함수이며 Modulation scalar 범위를 0 과 1 사이로 제한한다. B 는 Input feature map의 각 위치에 적용할 DCN [21]의 Kernel size를 가리키며, 3×3 Kernel 기반 DCN 연산을 사용하는 MDANet의 경우 9에 해당한다. 그림 19는 Multi-scale BEV feature 집합을 통해 예측된 Deformable mask가 \mathbf{F}_{t-n} 을 정렬하기 위해 DCN [21]연산에 사용되는 과정을 묘사하고 있다.

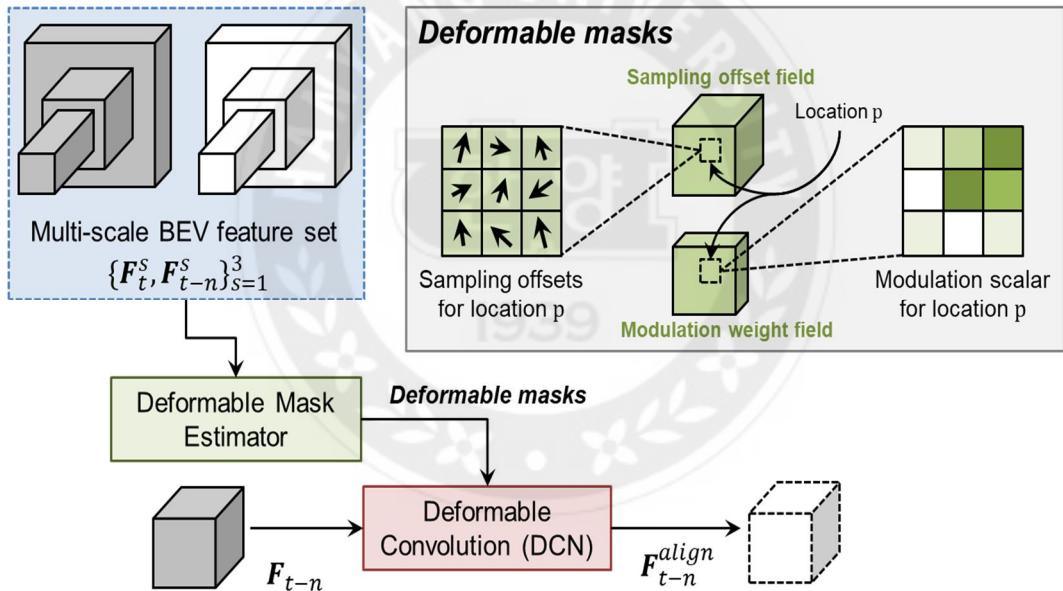


그림 19. $t - n$ 번째 프레임의 BEV feature map에 대한 특징 정렬 과정

\mathbf{F}_{t-n} 에 대한 임의의 2 차원 위치 좌표를 $p (= (p_x, p_y))$ 라 할 때, 위치 p 에 대한 DCN [21] 연산 과정은 식 (11)과 같다.

$$\mathbf{F}_{t-n}^{align}(p) = \sum_{b=1}^B W_b \cdot \mathbf{F}_{t-n}(p + \hat{p}_b + \Delta_{t-n}^b(p)) \cdot W_{mod,t-n}^b(p) \quad (11)$$

W_b 는 해당 Kernel의 b 번째 Learnable weight를 가리키며, $\hat{p}_b \in \{-1, -1\}, (-1, 0), \dots, (0, 1), (1, 1)\}$ 는 Kernel에 대해 사전 정의된 b 번째 Sampling

offset 을 가리킨다. $\Delta_{t-n}(p)$ ²⁰ 는 Sampling offset field Δ_{t-n} 의 위치 p 에 대한 Vector 를 가리키며, $\Delta_{t-n}^b(p)$ 는 해당 Vector 의 b 번째 원소를 가리킨다. 마찬가지로 $W_{mod,t-n}(p)$ ²¹는 Modulation scalar field $W_{mod,t-n}$ 의 위치 p 에 대한 값을 가리키며, $W_{mod,t-n}^b(p)$ 는 해당 Vector 의 b 번째 원소를 가리킨다. 또한 $p + \hat{p}_b + \Delta_{t-n}^b(p)$ 는 분수 값이므로 입력 Feature map 에 해당하는 \mathbf{F}_{t-n} 에서 Bilinear interpolation 을 통해 특징 값을 Sampling 한다.

($N - 1$) 개 Support 프레임에 대한 BEV feature map $\{\mathbf{F}_n\}_{n=t-N+1}^{t-1}$ 에 제안하는 MDANet 을 적용한 결과, Target 프레임의 BEV 공간에 맞게 정렬됨과 동시에 \mathbf{F}_t 의 정보량을 강화하는데 필요한 $\{\mathbf{F}_n^{align}\}_{n=t-N+1}^{t-1}$ 를 얻는다.



²⁰ $\Delta_{t-n}(p) = [\Delta_{t-n}^1(p), \dots, \Delta_{t-n}^b(p), \dots, \Delta_{t-n}^B(p)] \in \mathbb{R}^{2B}$, $\Delta_{t-n}^b(p) = [\Delta_{x,t-n}^b(p), \Delta_{y,t-n}^b(p)] \in \mathbb{R}^2$

²¹ $W_{mod,t-n}(p) = [W_{mod,t-n}^1(p), \dots, W_{mod,t-n}^b(p), \dots, W_{mod,t-n}^B(p)] \in \mathbb{R}^B$, $W_{mod,t-n}^b(p) \in \mathbb{R}$

4.3 Feature Aggregation by Alignment

Target 프레임에 대한 Refined BEV feature map \mathbf{F}'_t 을 도출하기 위해 제 4 장에서 제안하는 Long-term BEV feature refinement 알고리즘의 구조는 그림 20 에 묘사되어 있다. N 프레임 입력에 대해 해당 장기 시퀀스 파이프라인은 MDANet 을 통해 특징 정렬을 수행하여 $\{\mathbf{F}_n^{align}\}_{n=t-N+1}^{t-1}$ 를 얻은 뒤, 이를 4.1 절 선행연구에서 소개한 특징 결합방식²²을 통해 \mathbf{F}_t 와 결합한다. 장기 시퀀스 시간 영역에 대해 BEV 공간상에서 정보량이 강화된 Target 프레임의 Feature map 은 Detection head 로 전달되어 3 차원 객체에 대한 Classification 과 Localization 을 수행한다.

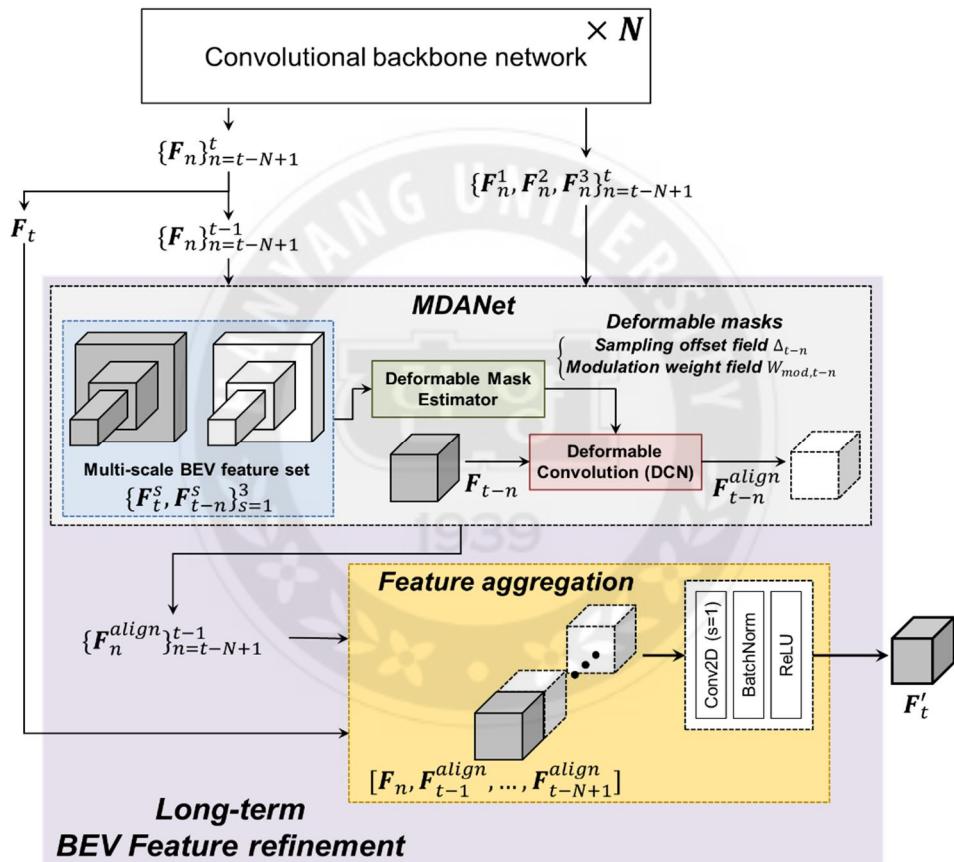


그림 20. Long-term BEV feature refinement 파이프라인 구조

제 4 장에서는 특징 정렬에 따른 결합을 수행하기 위해 MDANet 알고리즘을 제안했고, 이를 바탕으로 장기 시퀀스 관점에서 BEV 공간에 대한 시공간 특징 정보를 도출하는 파이프라인을 최종적으로 제안한다.

²² 식 (6) 참조

제5장 검증 실험

5.1 nuScenes 데이터셋

대표적인 도심 자율주행 데이터셋 중 하나인 nuScenes 데이터셋은 모두 1000개의 Scene이 제공되며 학습 용도로 600 Scenes, 검증 용도로 150 Scenes, 그리고 Benchmark 용도로 150 Scenes이 제공된다. 각 Scene은 약 20초 분량의 시퀀스 데이터로 구성되어 있으며 라이다 센서와 레이더 센서에 대한 포인트 클라우드 데이터와 카메라 센서에 대한 이미지 데이터를 함께 제공한다. 특히 라이다 센서의 경우, 32채널 회전식 라이다 센서를 통해 20Hz로 취득된 360° 범위 포인트 클라우드가 0.05초 간격을 두고 시퀀스 데이터 형태로 제공된다. 3차원 객체 검출에 관한 Ground truth(GT) 정보는 직육면체(Cuboid) 형태로 2Hz 마다 모두 10개 클래스²³에 대해 Annotation 되어 제공된다. 이때, Annotation 정보가 제공되는 포인트 클라우드 데이터를 Key sample이라 하며 포인트 클라우드 시퀀스 데이터에서 0.5초 간격으로 제공된다.

3 차원 객체 검출 성능에 관한 주요 지표는 mAP 와 NDS, 두 가지가 있다. mAP 성능의 경우, GT 데이터와 검출 결과 사이의 IoU²⁴ 값을 기반으로 측정하는 KITTI 데이터셋 [25]과 달리 중심 좌표 간 BEV 공간상 2 차원 직선 거리를 바탕으로 Thresholding 을 적용해 측정한다. NDS 성능의 경우, mAP 이외의 5 가지 검출 성능 항목²⁵을 mAP 와 함께 가중치 합으로 종합하여 얻는다.

5.2 포인트 클라우드 시퀀스 기반 알고리즘 실험 구성

포인트 클라우드 시퀀스 데이터의 시공간 표현 학습을 위해 본 연구에서 제안된 알고리즘에 대한 검증 실험은 PointPillars [2]와 CenterPoint [26]를 3 차원 객체 검출 Baseline 모델로 삼아 진행했다.

포인트 클라우드 데이터 처리 및 모델 구현

객체 검출에 사용하는 단일 프레임의 포인트 클라우드 입력은 nuScenes 데이터셋에서 제공되는 Key sample 마다 10 sweep 단위로 병합을 통해 구성한다. 따라서 단일 프레임 입력은 0.5 초 분량의 단기 시퀀스 데이터가 병합된 포인트

²³ 10 가지 클래스: Car, truck, construction vehicle, bus, trailer, barrier, motorcycle, bicycle, pedestrian, traffic cone

²⁴ Intersection over union

²⁵ mean average translation error(mATE), mean average scale error(mASE), average orientation error(mAOE), mean average velocity error(mAVE), mean average attribute error(mAAE)

클라우드 데이터²⁶에 해당한다. 또한 모델의 입력으로 사용하는 3 차원 포인트 클라우드 데이터의 공간 범위는 라이다 좌표계의 (x, y, z) 축에 대해 PointPillars [2]의 경우, $[-51.2, 51.2] \times [-51.2, 51.2] \times [-5.0, 3.0] m$ 를 사용했고 CenterPoint [26]의 경우, $[-54, 54] \times [-54, 54] \times [-5.0, 3.0] m$ 를 사용했다. 이후 소개할 검증 실험 과정에서 별도의 언급이 없다면 각각의 포인트에 대한 초기 특징 벡터는 3.3 절에서 소개한 종래 방식에 맞게 구성했다²⁷.

모델 구현

3 차원 포인트 클라우드 입력을 격자 형태로 구분하여 특징 표현을 얻기 위해 PointPillars [2]의 경우, 크기가 $(0.2 \times 0.2 \times 8.0) m$ 인 Pillar 를 사용하고 CenterPoint 의 경우, 크기가 $(0.075 \times 0.075 \times 0.2) m$ 인 Voxel 을 사용했다. 단기 시퀀스 알고리즘에 해당하는 SA-GFE 의 효용성을 확인하기 위한 검증 실험의 경우, PointPillars [2]의 Pillar feature encoding 와 CenterPoint [26]의 Average pooling 를 각각 SA-GFE 로 대체하여 단기 시퀀스 알고리즘이 적용된 단일 프레임 입력 기반 검출 모델을 구현했다. 또한 장기 시퀀스 관점 알고리즘 MDANet 을 포함하고 있는 Long-term BEV feature refinement 파이프라인의 효용성을 확인하기 위해, Convolutional backbone network 와 Detection head 사이에 해당 파이프라인을 추가하여 다중 프레임 입력 기반 모델로 확장하여 구현했다. 본 논문에서 제안하는 두 가지 알고리즘이 모두 적용된 최종 모델, LSR-3D 는 SA-GFE 를 적용한 PointPillars 모델을 다중 프레임 입력 기반 파이프라인으로 확장한 뒤, Convolution backbone network 와 Detection head 중간에 Long-term BEV feature refinement 파이프라인을 적용하여 구현했다.

지도 학습 및 평가

검증 실험을 위한 모델에 대한 지도 학습은 SA-GFE 알고리즘에 관한 단일 프레임 기반 모델 학습과 Long-term BEV feature refinement 파이프라인이 적용된 다중 프레임 기반 모델 학습으로 구분된다. 다중 프레임 기반 모델의 경우, 해당 모델의 단일 프레임 기반 파이프라인을 사전 학습(Pretraining) 시킨 후²⁸, 조정 학습(Fine-tuning)을 통해 전체 파이프라인을 학습하는 방식을 사용했다. 검출 결과를 바탕으로 한 손실 함수(Loss function) 계산은 Baseline 모델에서 사용한 손실 함수를 동일하게 사용했다. 다시 말해 PointPillars [2]에 알고리즘을 적용한

²⁶ 2.2절 다중 스윕 전처리 방식 참조

²⁷ 3.3절에서 소개한 Temporal-bin based point feature initialization을 적용하지 않은 경우, 그림 11 참조

²⁸ PointPillars 모델에 SA-GFE 알고리즘과 Long-term BEV feature refinement 알고리즘을 모두 적용한 다중 프레임 기반 모델의 경우, SA-GFE가 적용된 PointPillars 모델을 단일 프레임 기반 모델 학습 방식에 맞춰 사전 학습 진행.

경우, Anchor head 기반 손실 함수를 사용하고 CenterPoint [26]에 알고리즘을 적용한 경우, Anchor free head 기반 손실 함수를 적용하였다. 손실 값을 이용한 최적화는 모든 실험에 대해 Adam optimizer를 사용했다. 또한 학습률(Learning rate) scheduling 은 One cycle learning rate scheduling 방식을 사용했다. 이때 단일 프레임 기반 모델 학습의 경우, 최대 LR²⁹ 값을 0.001로 설정했으며, 다중 프레임 기반 모델, 즉 장기 시퀀스 파이프라인이 적용된 모델을 학습하는 경우에는 최대 LR 값을 0.0002로 설정했다. Data augmentation 의 경우, random world flipping, random world rotation, random world scaling 그리고 GT Sampling [27]을 적용했다.

제 5 장에서 진행된 모든 실험은 nuScenes 학습용 데이터³⁰의 7 분의 1 분량(Mini train set)만 사용하여 학습한 경우와 전체를 모두 사용해 학습한 경우로 나뉜다. 성능 평가는 항상 검증용 데이터셋³¹ 전체를 사용했으며, FPS 성능과 Memory usage 은 단일 GPU 환경에서 TITAN RTX를 통해 측정되었다.

5.3 SA-GFE 알고리즘 검증 실험

실험 5.3.1 SA-GFE 알고리즘 구성 요소에 대한 Ablation study

본 실험은 제안된 단기 시퀀스 알고리즘 요소에 따른 성능을 검증하였다. 알고리즘 효용성을 빠르게 비교하기 위해 Mini train set 으로 지도 학습을 진행했다. 본 실험에서 사용한 단일 프레임 입력 기반 모델은 PointPillars [2]다. 따라서 해당 모델의 Pillar feature encoder 를 그대로 사용한 경우가 Baseline 에 해당한다. 먼저 TCANet 을 추가하여 성능을 평가함으로써 Pillar 공간에 포함된 시공간 표현을 학습하는 알고리즘의 효용성을 확인했다. 또한 시공간 표현 학습 효과를 높이기 위한 Temporal-bin 기반 Point feature augmentation 기법을 추가하여 성능을 비교함으로써 제안된 SA-GFE 알고리즘을 검증했다.

TCANet 을 Pillar feature encoding 과정에 추가한 경우, Baseline 대비 mAP 와 NDS 성능이 각각 1.58%, 1.34% 씩 향상되었음을 표 2에서 확인할 수 있다. 따라서 단기 시퀀스에 포함된 시공간 특징 표현을 학습하고 이를 활용할 때, 종래 방식보다 효과적으로 각각의 Pillar 공간에 대한 Encoding 이 가능함을 알 수 있다. 또한 Temporal bin에 기반해 초기 포인트 특징 벡터의 정보량을 늘린 다음 Pillar feature encoding 을 수행할 때, mAP 와 NDS 성능이 각각 0.7%, 0.52% 씩 추가적으로 향상되었다. 이를 통해 단기 시퀀스 범위를 더 작은 시구간 단위로 구분한 뒤 격자 공간 내 포인트들의 위치 관계 정보를 활용하는 Augmentation 방식의 효용성을 확인할 수 있다.

²⁹ LR: Learning rate

³⁰ nuScenes train set: 28130 samples

³¹ nuScenes validation set: 6019 samples

Pillar encoding method	mAP [%]	NDS [%]	FPS [frame]	Memory usage [MiB ^{32]}
Baseline	37.81	53.1	34	3608
+ TCANet	39.39(+1.58)	54.44(+1.34)	29.8	4144
+ Temporal Bin-Based Augmentation	40.09(+2.28)	54.96(+1.86)	28.7	4150

표 2. SA-GFE 알고리즘 구성 요소에 따른 성능 평가

실험 5.3.2 Pillar 기반 객체 검출 모델에 대한 SA-GFE 알고리즘의 효용성 평가

본 실험은 nuScenes 데이터셋에 대한 SA-GFE 알고리즘의 효용성을 확인하기 위해 진행한 실험이다. Baseline 은 실험 5.3.1 과 동일하게 PointPillars [2]를 사용했으며, 표 3 의 SA-GFE 는 Baseline 의 Pillar feature encoder 를 SA-GFE 로 대체한 모델³³이다. 두 가지 경우 모두 nuScenes 전체 학습 데이터셋을 이용해 학습 후, 성능을 평가했다.

Method	mAP [%]	NDS [%]
Baseline	44.74	58.32
SA-GFE	46.25(+1.51)	59.4(+1.08)

표 3. Pillar 기반 검출기에 대한 SA-GFE 효용성 평가 결과

병합된 포인트 클라우드 시퀀스에 대한 Pillar 공간 영역의 Feature encoding 방식을 SA-GFE 로 대체한 결과, mAP 와 NDS 성능이 각각 1.51%, 1.08% 씩 향상되었음을 표 3 에서 알 수 있다. 또한 클래스 별 mAP 성능이 정리된 표 4 를 통해 다음 두 가지를 확인하였다. 첫째, 단기 시퀀스 내 취득된 포인트 수가 차량 관련 클래스에 비해 상대적으로 적고 객체 형태 학습에 있어 Noise point 가 야기하는 Disturbance 에 취약한 객체 클래스(Pedestrian, Traffic cone)에서 높은 성능 향상을 보였다. 둘째, Pillar 공간에 포함된 포인트 수준에서 Attention 을 수행하는 SA-GFE 알고리즘을 적용한 결과, 일부 차량 클래스에서 성능 향상이 거의 없거나 오히려 성능 저하가 발생하였다. 이는 객체 크기가 상대적으로 크고 움직임이 많은 차량 관련 클래스의 경우, Localization 성능 개선을 위해 더 긴 시간 범위에 대한 시공간

³² MiB: Mega binary byte, 1MiB = 2^{20} byte

³³ 그림 7에 묘사된 검출기 전체 파이프라인에서 Pillar feature encoding 부분만 대체

표현 학습의 필요성을 시사한다. 다시 말해 Convolutional layer 를 통과한 BEV Feature map 과 같이 Feature map 의 각 위치마다 더 넓은 Receptive field 정보가 포함된 공간 영역에서 시공간 정보를 활용해야 하는 것이다.



Method	Car	Truck	C.V ³⁴	Bus	Traile r	Barrier	Motorcycle	Bicycle	Pedestria n	T.C ³⁵
Baseline	81.28	50.01	12	63.64	35.43	49.92	29.39	6.33	72.33	47.07
SA-GFE	81.2	50.27	11.45	64.13	34.49	50.38	35.06	9.52	75.39	50.64

표 4. Pillar 기반 검출기에 대한 SA-GFE 효용성 평가 실험 클래스별 mAP 성능

³⁴ Construction vehicle

³⁵ Traffic cone

실험 5.3.3 Voxel 기반 객체 검출 모델에 대한 SA-GFE 알고리즘의 효용성 평가

본 실험은 격자 단위로 특징 벡터를 Encoding 할 때, SA-GFE 알고리즘이 Pillar 공간보다 더 작은 Voxel 공간에 대해서 어떤 효용성을 갖는지 확인하기 위해 진행되었다. Baseline은 CenterPoint [26]를 사용했고, 그림 21에서 묘사하듯 SA-GFE는 CenterPoint의 Convolutional backbone 파이프라인의 앞 단계에 해당하는 Voxel 별 Average pooling 을 대체하여 적용되었다. 또한 nuScenes 학습용 전체 데이터셋을 가지고 학습한 뒤 성능을 평가했다.

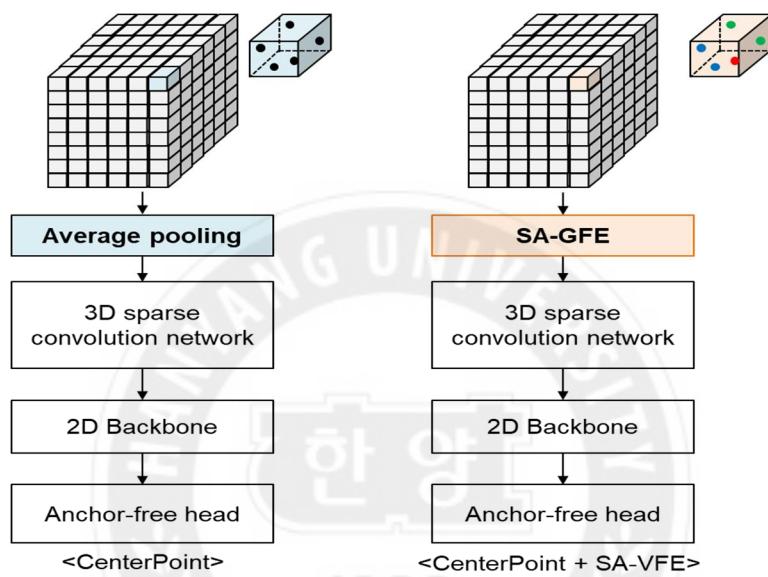


그림 21. Voxel 기반 객체 검출 모델에 대한 SA-GFE 적용 유무에 따른 파이프라인

<i>Method</i>	mAP [%]	NDS [%]	mATE [m]	mASE [1-IOU]	mAOE [rad]	mAVE [m/s]	mAAE [1-acc]
Baseline	57.9	65.21	0.304	0.261	0.387	0.237	0.183
SA-GFE	57.87	65.82	0.296	0.257	0.358	0.213	0.187

표 5. CenterPoint에 SA-GFE를 적용한 실험 결과

Voxel 공간의 특징 벡터 도출 방식을 SA-GFE로 대체한 결과, mAP 성능은 0.03% 감소했고 NDS 성능은 0.61% 향상되었음을 표 5에서 확인할 수 있다. 또한 mAAE 를 제외한 나머지 검출 오류 지표는 Baseline 대비 개선되었다. 이를 통해 Voxel 공간에 대한 SA-GFE 알고리즘의 효용성을 확인했다. 하지만 Pillar 공간에서 SA-GFE를 적용한 경우보다 성능 향상 폭이 낮고 특히 mAP 성능은 거의 증감을 보이지 않았다. 상기 결과의 요인은 표 6을 통해 분석 가능하다.

격자 형태	Pillar (0.2 × 0.2 × 8.0) m	Voxel (0.75 × 0.75 × 0.2) m
격자에 포함된 평균 포인트 개수	7.67 개	2.89 개
Non-empty 격자 중 지정된 최대 포인트 개수 ³⁶ 만큼 포인트가 담긴 격자 비율	17.9 %	8.63 %
Non-empty 격자 중 지정된 10 개 이상 포인트를 포함한 격자 비율	31.96 %	8.63 %

표 6. 격자 형태에 따른 nuScenes 검증용 포인트 클라우드 데이터의 통계표

표 6 은 nuScenes 검증용 데이터셋에 대해 10sweep 단위로 단일 프레임을 구성해 실험 5.3.2 와 5.3.3 에서 각각 사용한 Pillar 와 Voxel 형태로 3 차원 공간에 대한 복셀화를 수행한 결과 집계된 통계표다. Baseline 으로 삼은 PointPillars [2] 와 CenterPoint [26]에서 사용한 격자 크기에 따라 격자 별 포인트 밀도 차이가 큰 것을 알 수 있다. 특히 SA-GFE 알고리즘은 격자에 담긴 10 sweep 단위 포인트 시퀀스에 대해 각 시점 모두를 고려하여 Attention weight 를 계산한다. 따라서 Pillar 보다 훨씬 작은 공간의 Voxel 에 적용한 실험 5.3.3 의 경우, 평균 포인트 수가 3 개 미만인 Voxel 구조로부터 TCANet 알고리즘의 효과를 얻기 힘들다. 반면 단기 시퀀스 데이터의 Voxel 공간에 대한 시공간 표현 학습 효과는 Voxel 공간의 크기가 크게 설정될수록 그 효과가 두드러지는 것으로 해석 가능하다. 모델의 Voxel 크기 설정이 연산 처리 성능과 검출 정확도 성능에 대한 Trade-off 를 고려하여 정해진다는 점을 고려할 때, SA-GFE 는 Voxel 크기를 증가로 인한 검출 정확도 감소를 낮출 수 있다는 효용성을 갖는다.

³⁶ Hard voxelization 수행에 필요한 Hyperparameter. PointPillars는 20개, CenterPoint는 10개로 설정하여 실험 진행

5.4 Long-term BEV feature refinement 알고리즘 검증 실험

실험 5.4.1 다중 프레임 입력 길이에 따른 성능 검증

본 실험은 제안된 BEV feature refinement 파이프라인의 효용성을 검증하기 위해 장기 시퀀스 입력 길이, 즉 모델이 사용한 다중 프레임 입력 수에 따른 성능을 평가했다. Baseline 모델로 PointPillars [2]를 사용했고 사전 학습은 학습용 데이터셋 전체를 사용³⁷했다. Long-term BEV feature refinement 파이프라인을 포함시켜 다중 프레임 입력에 대한 검출 모델 학습을 학습하는 조정 학습 단계는, 알고리즘 효용성을 빠르게 비교하기 위해 Mini train set으로 학습했다.

	Baseline	<i>Long-term BEV Feature Refinement</i>		
Sequence length [frame]	1	2	3	4
mAP [%]	44.4	47.2(+2.8)	47.65(+3.25)	48.04(+3.64)
NDS [%]	58.15	59.85(+1.7)	60(+1.85)	60.2(+2.05)
FPS [frame]	34	21.5	17.2	14.5

표 7. 다중 프레임 입력 길이에 따른 검출 성능

4.1 절 선행 연구를 통해 장기 시퀀스 범위의 BEV 공간 영역에서 Spatial feature displacement 를 고려하지 않고 여러 BEV feature map 을 결합하는 경우, 다중 프레임 입력 길이가 늘어남에 따른 성능이 저하되는 것을 표 1 에서 확인했다. 반면, 본 연구에서 제안하는 MDANet 을 통해 특징 정렬을 먼저 수행한 뒤, 특징 결합을 수행하는 경우, 입력 프레임 수가 늘어날수록 검출 성능이 향상되는 것을 표 7 에서 확인할 수 있다. 따라서 인접한 프레임간 발생하는 Spatial feature displacement 문제를 MDANet 기반 특징 정렬 알고리즘을 통해 개선할 수 있는 것으로 분석 가능하다. 또한 이를 통해 본 연구에서 제안하는 Feature aggregation by alignment 방식 기반의 장기 시퀀스 알고리즘이 갖는 효용성을 확인할 수 있다.

³⁷ Pretraining과 Fine-tuning 모두 Mini-batch 크기 4

Sequence length [frame]	2	3	4
Model runtime [ms]	46.51	58.14	68.97
Long-term pipeline runtime [ms]	18.2	30.15	41.25
Long-term pipeline runtime ratio [%]	39.13	51.86	59.81

표 8. 다중 프레임 입력 길이에 따른 장기 시퀀스 파이프라인의 Runtime 비율

반면 Feature refinement 과정에서 사용하는 시퀀스 길이가 늘어남에 따라 장기 시퀀스 파이프라인이 전체 검출 파이프라인에서 차지하는 Runtime 비중이 늘어나는 것을 표 8에서 확인 가능하다. 따라서 실제 임베디드 환경에서 구현 시, 검출 정확도 성능과 연산 처리 성능의 Trade-off 를 고려해 장기 시퀀스 입력 길이를 정하는 것이 중요하며 H/W 최적화를 통해 해당 파이프라인의 Runtime 을 줄이는 것도 고려할 수 있다.

실험 5.4.2 MDANet의 Deformable mask estimator 설계 방식 검증 실험

설계 방식에 대한 첫 번째 검증 실험은 Deformable mask estimator 가 Backbone network 연산 결과를 활용하는 방식³⁸의 효용성에 대해 검증한다. 이를 위해 Multi-scale feature 가 하나의 Scale로 통합된 $\{F_n\}_{n=t-N+1}^t$ 을 이용해 Deformable mask 를 예측하는 설계 방식을 MDANet 에 적용하여 성능을 비교했다. 그림 22 는 본 실험을 위해 추가로 설계된 방식이다. Backbone network 에서 직접 전파된 Multi-scale feature 를 사용하는 그림 18 과 달리 하나의 Feature scale에서 다시 Multi-scale motion feature 를 도출한다.

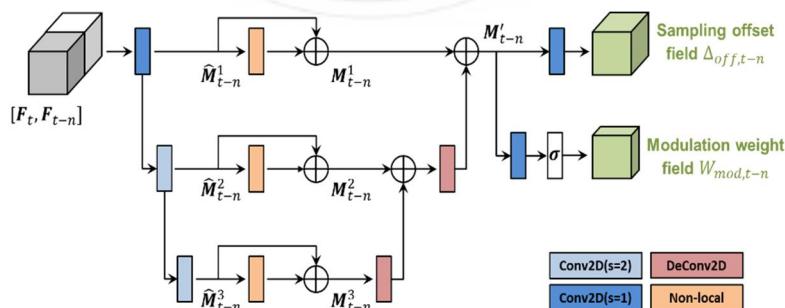


그림 22. 단일 스케일 BEV feature map 기반 Deformable mask estimator

³⁸ Motion feature 도출을 위해 Backbone network 연산 과정에서 전달된 Multi-scale BEV feature를 직접 이용한다. 4.2절 참고.

설계 방식에 대한 두 번째 검증 실험은 Motion feature 도출 과정에서 Non-Local block [24]을 포함시키는 것이 갖는 효용성을 검증한다. 이를 위해 Deformable mask estimator 구조에서 Non-local block [24]이 제외된 구조를 추가로 설계했다. 그림 23 은 Backbone network 의 Multi-scale BEV feature 를 직접 사용하는 대신 Non-local block [24]이 제외된 Deformable mask estimator 구조를 보여준다.

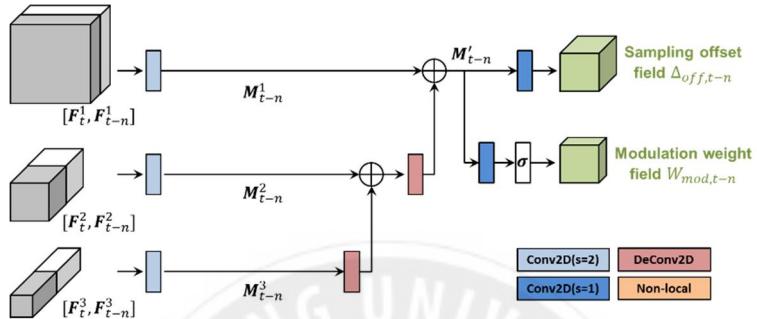


그림 23. Non-local block 을 제외시킨 Deformable mask estimator

	Design methodology		mAP [%]	NDS [%]	FPS [frame]	Memory usage [MiB]
	Source feature	NL Block ³⁹				
Baseline			44.4	58.15	34	3608
Deformable mask estimator	SS ⁴⁰ feature	✓	46.59(+2.19)	59.44(+1.29)	17.5	7242
	MS ⁴¹ feature		46.72(+2.32)	59.55(+1.4)	19.2	6592
	MS feature	✓	47.65(+3.25)	60(+1.85)	17.2	7624

표 9. Motion feature 도출에 관한 설계 방식에 따른 성능

표 9 는 MDANet 의 Deformable mask estimator 를 각각 그림 22 과 그림 23 에서 묘사한 구조로 대체한 경우와 본 논문에서 제안하는 구조에 대한 경우의

³⁹ NL Block: Non-local block [16]

⁴⁰ SS: Single-scale

⁴¹ MS: Multi-scale

객체 검출 성능을 순서대로 제시하고 있다. Baseline 은 단일 프레임 기반 PointPillars [2]에 해당하며, 장기 시퀀스 파이프라인을 적용한 3 가지 경우 모두 포인트 클라우드 시퀀스 입력으로 Target 프레임을 포함해 연속된 3 개 프레임을 사용했다. Deformable mask estimator 의 Source feature 로 Single-scale feature 를 이용하는 경우와 4.2 절에서 제안된 구조의 경우를 비교해볼 때, Multi-scale feature 를 사용할 때, Feature scale 에 따른 정보 손실을 최소화한 상태에서 효과적인 Motion feature 를 도출이 가능함을 알 수 있다. 또한 FPS 성능은 거의 차이가 발생하지 않았는데 이는 Multi-scale feature map 을 Source feature 로 사용하는데 있어 별도의 파이프라인이 필요하지 않기 때문⁴²이다. 또한 Single-scale feature 를 사용한 경우와 Multi-scale feature 를 사용하지만 Non-local block 연산을 사용하지 않은 경우에 대한 비교를 통해 Non-local block [24]을 추가해 넓은 범위의 Spatial 정보를 활용하는 것보다 Multi-scale feature map 을 직접 활용하여 Motion feature 를 도출하는 것이 BEV 공간에서 Deformable mask 를 예측하는데 있어 더 중요한 설계 요소임을 알 수 있다.

실험 5.4.3 Anchor-free detector에 대한 Long-term BEV feature refinement 효용성 검증

본 실험은 제안된 장기 시퀀스 알고리즘을 Anchor-free 기반 Detection head 를 사용하는 검출 모델에 적용함으로써 제안된 알고리즘이 종래 다양한 격자 기반 검출 모델들에 적용될 수 있음을 확인한다. 따라서 Baseline 모델로 CenterPoint [26]를 사용했다. 사전 학습과 조정 학습 모두 학습용 데이터셋 전체를 사용⁴³ 했다. 장기 시퀀스 알고리즘이 적용된 CenterPoint 기반 검출 모델(Long-term model)은 포인트 클라우드 시퀀스 입력으로 Target 프레임 포함 3 개의 프레임을 사용했다.

<i>Model</i>	Sequence length [frame]	mAP [%]	NDS [%]
Baseline	1	59.22	66.48
Long-term model	3	61.44(+2.22)	67.47(+0.99)

표 10. Anchor-free 기반 detector 에 대한 장기 시퀀스 파이프라인 검증

종래 Anchor-free detector 를 다중 프레임 입력 기반 파이프라인으로 확장해 시공간 특징 표현을 활용한 결과, Anchor-based detector 의 경우⁴⁴와 마찬가지로,

⁴² 4.2절 참고

⁴³ Pretraining과 Fine-tuning 모두 Mini-batch 크기는 8

⁴⁴ 표 7 참고

검출 성능이 향상되었음을 표 10에서 확인할 수 있다. 따라서 본 연구에서 제안하는 Target 프레임 BEV representation 정보량 강화 방식은 Detection head의 종류와 무관하게 효용성을 보였다.

5.5 LSR-3D 모델 검증 실험

계층적 관점에 기반한 시공간 특징 표현 학습 모델 성능 검증

본 실험에서는 단기 시퀀스 관점에 관한 SA-GFE 알고리즘과 장기 시퀀스 관점에 관한 BEV feature refinement 알고리즘이 모두 적용된 통합 모델에 해당하는 LSR-3D의 객체 검출 성능을 평가했다. 이를 통해 계층적 관점 기반 시공간 특징 표현 학습 방식의 효용성을 검증한다. Baseline은 PointPillars [2]를 사용했다. 장기 시퀀스 알고리즘을 적용하지 않은 경우는 단일 프레임 입력을 사용했으며, 적용한 경우는 다중 프레임 입력으로 3개 프레임⁴⁵을 사용해 실험을 진행했다. 3개 프레임 입력에 대한 LSR-3D의 전체 파이프라인 구조는 그림 24에 제시되어 있다.

각 경우에 대한 학습 방법을 정리하면 다음과 같다. 먼저 Baseline의 경우와 SA-GFE 알고리즘만 적용된 경우 각각은 단일 프레임 기반 검출 모델을 학습용 데이터셋 전체를 이용해 사전 학습한 뒤, 성능을 확인하였다. 또한 각각의 모델 Weight를 다중 프레임 기반 검출 모델의 경우에 대한 초기 Weight로 삼고 학습용 데이터셋 전체를 이용해 조정 학습을 진행했다⁴⁶.

Model	<i>Proposed Method</i>		mAP [%]	NDS [%]	FPS [frame]
	SA-GFE	Long-term BEV Feature Refinement			
Baseline			44.63	58.36	34
LSR-3D	✓		46.25(+1.62)	59.4(+1.04)	28.7
		✓	48.76(+4.13)	60.26(+1.9)	17.2
	✓	✓	50.17(+5.54)	61.43(+3.07)	14.5

표 11. 통합 모델 LSR-3D에 대한 Ablation study

단기 시퀀스와 장기 시퀀스, 두 가지 관점의 시공간 특징 표현 알고리즘을 모두 적용한 통합 모델은 Baseline 대비 mAP와 NDS 각각 5.54%와 3.07% 씩 향상된

⁴⁵ 2개의 Support 프레임과 1개의 Target 프레임

⁴⁶ Pretraining과 Fine-tuning 각각 Mini-batch 크기는 16, 8

검출 성능을 보였다. 표 12, 13 를 통해 단기 시퀀스 알고리즘이 Baseline 에 적용되었을 때, 동일한 단일 프레임 입력을 통해 Pedestrian 과 Traffic cone 클래스에서 높은 성능 향상 폭을 보인 것을 알 수 있다. 한편, Baseline 에 대해 장기 시퀀스 알고리즘을 적용했을 때, 인접한 프레임 사이에서 움직임을 주로 보이지만 단일 프레임만 사용할 경우 검출이 어려운 동적 객체 관련 클래스⁴⁷와 종횡비가 큰 차량 클래스⁴⁸에서 성능이 많이 향상됨을 확인할 수 있다. 또한 두 가지 관점의 알고리즘이 모두 적용되어 통합 모델을 구성했을 때, Baseline 에서도 검출 성능이 상대적으로 높았던 Car 클래스를 제외하고 모든 객체 클래스에서 높은 성능 향상을 보인 것을 확인할 수 있다.

표 11 에 제시된 연산 처리 성능을 비교해보면 단일 프레임 입력을 사용하는 Baseline 의 경우 1 초에 34 프레임을 처리할 수 있는 연산 성능을 보였고 3 개의 연속된 프레임을 사용하는 통합 모델의 경우, 1 초에 14.5 프레임에 대한 검출 결과 예측이 가능한 성능을 보였다. 통합 모델이 Baseline 대비 3 배 분량의 입력 시퀀스 데이터를 활용했음에도 불구하고 연산 처리 성능은 약 2.34 배 감소했다. 본 연구에서는 Inference 파이프라인에 대해 과거 프레임의 BEV feature map 을 저장해두는 Queue 형태의 Feature map memory bank 를 활용할 수 있도록 설계했다. 따라서 매 프레임 입력에 대해 해당 프레임의 BEV feature refinement 를 수행할 때, 과거 프레임에 관한 BEV feature map 은 새로 계산하지 않고 Memory bank 에서 불러와 사용 가능하도록 했으며 FIFO 방식으로 저장된 Feature map 을 업데이트하도록 설계함으로써 효과적인 장기 시퀀스 파이프라인 연산이 가능하도록 했다.

Model	<i>Proposed method</i>		Car	Truck	C.V ⁴⁹	Bus	Trailer
	SA-GFE	Long-term BEV Feature Refinement					
Baseline			81.56	49.99	10.63	61.63	33.05
LSR-3D	✓		81.2	50.27	11.45	64.13	34.49
		✓	81.9	51.6	16.9	65.4	36.5
	✓	✓	81.8	51.97	15.53	65.52	36.3

표 12. LSR-3D Ablation study 에 대한 클래스 별 mAP 성능 (1)

⁴⁷ Motorcycle, Bicycle

⁴⁸ Construction vehicle, Trailer

⁴⁹ Construction vehicle

Model	<i>Proposed method</i>		Barrier	Motor. ⁵⁰	Bicycle	Ped. ⁵¹	T.C ⁵²
	SA-GFE	Long-term BEV Feature Refinement					
Baseline			47.77	33.35	8.92	73.26	46.16
LSR-3D	✓		50.38	35.06	9.52	75.39	50.64
		✓	49.8	46.1	14.1	73.9	51.4
	✓	✓	50.92	50.44	17.52	76.25	55.45

표 13. LSR-3D Ablation study에 대한 클래스 별 mAP 성능 (2)



⁵⁰ Motorcycle

⁵¹ Pedestrian

⁵² Traffic cone

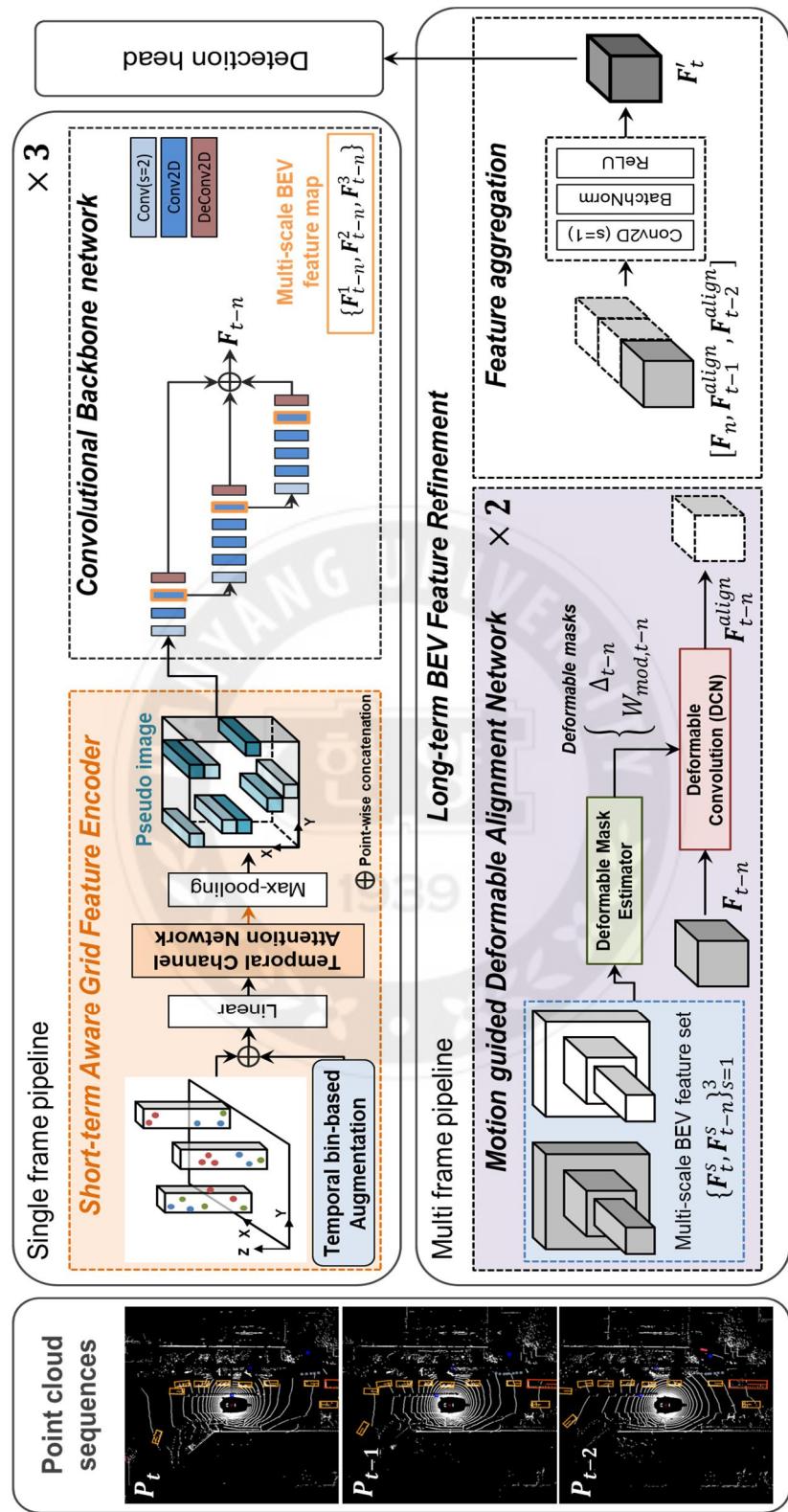


그림 24. LSR-3D의 전체 파이프라인 구조

정성적 분석을 통한 LSR-3D 모델 성능 검증

본 연구에서는 단기 시퀀스 알고리즘과 장기 시퀀스 알고리즘이 모두 적용된 LSR-3D 모델의 3 차원 객체 검출 성능을 주로 3 가지 경우에 대해 정성적으로 분석했다. Baseline은 PointPillar [2]를 사용했으며, LSR-3D 모델의 입력은 Target 프레임을 포함해 연속된 3 개 프레임을 입력으로 사용했다. nuScenes 데이터셋의 경우 모두 6 개의 카메라 채널⁵³에 대한 이미지 데이터를 제공하는데, 정성적 분석을 위해 라이다 기반 3 차원 객체 검출 결과를 확인하고자 하는 카메라 채널 이미지로 투영시켜 비교하였다.

첫 번째 경우는 ‘False positive 검출 감소’에 관한 분석이다. 그림 25~27은 특정 Scene⁵⁴에 대한 첫 번째부터 세 번째 프레임에 대한 검출 결과를 보여준다. 첫 번째 프레임(그림 25)의 경우, LSR-3D 모델 역시 과거 Support 프레임에 대한 BEV feature map 정보가 없으므로 Baseline과 마찬가지로 False positive 검출 비율이 높은 걸 확인할 수 있다. LSR-3D의 경우가 좀 더 적은 이유는 SA-GFE 알고리즘을 통해 Baseline의 경우보다 좋은 격자 별 특징 벡터를 도출하기 때문으로 분석 가능하다. 두 번째 프레임부터 장기 시퀀스 알고리즘이 효과를 내며 False positive 검출 비율이 대폭 줄어든 것을 알 수 있다.

두 번째 경우는 ‘포인트의 부분적 취득 문제 개선’에 관한 분석이다. 그림 27~29는 특정 Scene⁵⁵의 연속된 3 개 프레임에 대한 검출 결과를 각각 순서대로 보여준다. BEV에서 시각화한 결과를 보면 자율 이동체 기준 11 시 방향에 있는 차량 관련 객체들에 대해 나무와 도로 표지판 등에 가려 해당 객체 외형의 매우 일부분에 관한 포인트만 취득된 것을 알 수 있다. 그림 28~30을 통해 계층적 관점에 기반해 Baseline 보다 긴 시간에 걸친 포인트 클라우드 시퀀스 데이터를 활용함으로써 부분적 취득 문제가 개선된 것을 알 수 있다.

마지막 경우는 ‘중첩 상황 검출 성능 개선’에 관한 분석이다. 그림 31~33은 특정 Scene⁵⁶의 연속된 3 개 프레임에 대한 검출 결과를 각각 순서대로 보여준다. 단일 프레임 입력을 사용하는 Baseline의 경우, 인접한 프레임에서 얻은 특징 정보가 서로 독립적이므로 중첩이 발생한 경우에 대해 검출을 제대로 하지 못하는 것을 확인 가능하다. 반면 장기 시퀀스 알고리즘을 통해 시공간 맥락을 반영하여 Target 프레임에 대한 BEV feature refinement를 수행함으로써 풀숲으로 인한 중첩이 심한 객체에 대해서도 검출하는 걸 확인 가능하다. 또한 그림 34~36에 제시된 연속된 3 개 프레임에 대한 검출 결과 비교를 통해 중첩 상황에 대한 검출 성능이 개선된 것을 추가로 확인할 수 있다.

⁵³ Camera_front, camera_front_left, camera_front_right, camera_back, camera_back_left, camera_back_right

⁵⁴ nuScenes dataset scene token: 6776b91389394ff18e57b269863b4dbf

⁵⁵ nuScenes dataset scene token: 16e50a63b809463099cb4c378fe0641e

⁵⁶ nuScenes dataset scene token: e60ef590e3614187b7800db3e5284e1a



그림 25. False positive 검출 감소에 관한 첫 번째 프레임의 검출 결과



그림 26. False positive 검출 감소에 관한 두 번째 프레임의 검출 결과



그림 27. False positive 검출 감소에 관한 세 번째 프레임의 검출 결과

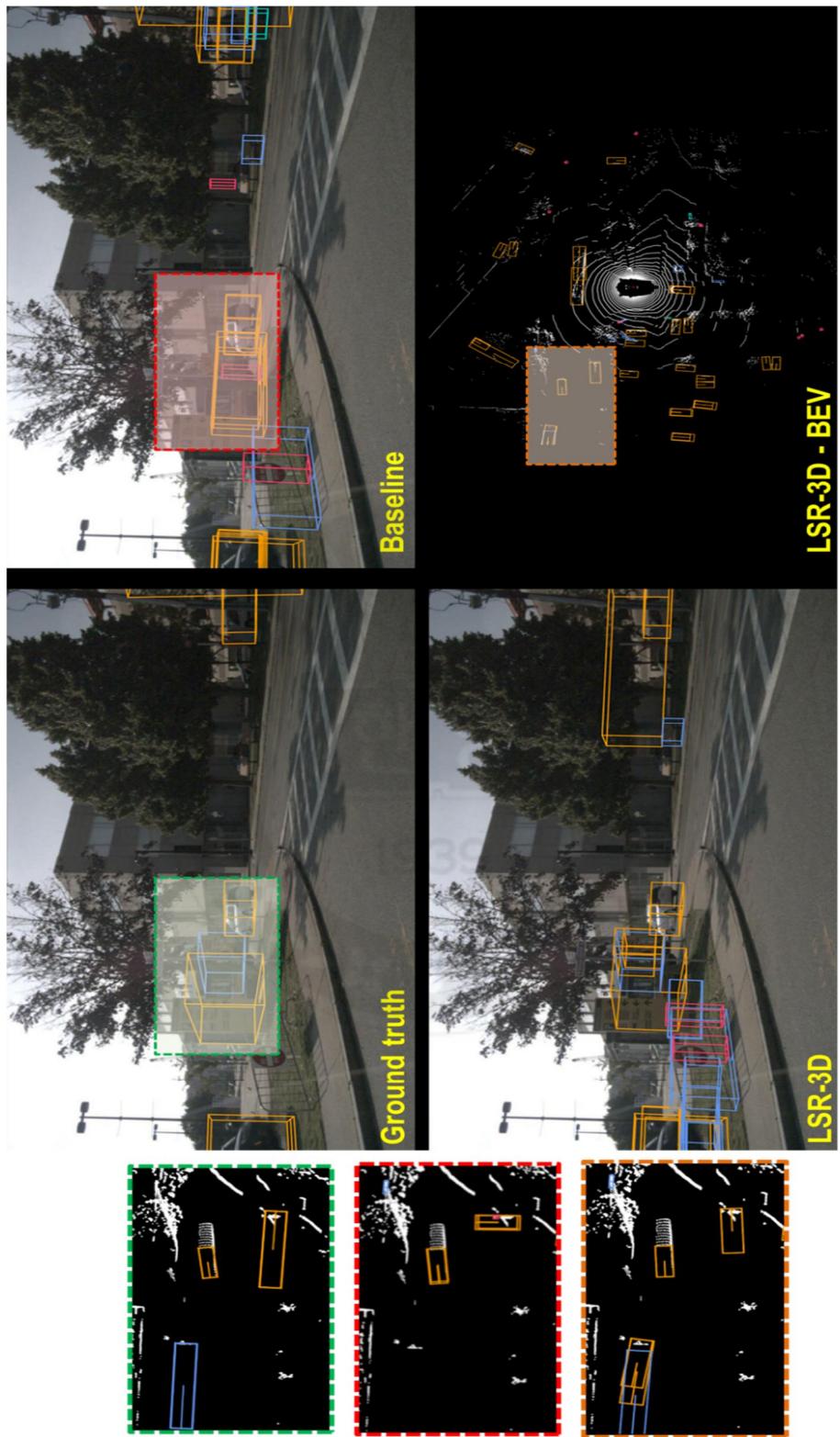


그림 28. Partial view problem 개선에 관한 첫 번째 프레임의 검출 결과

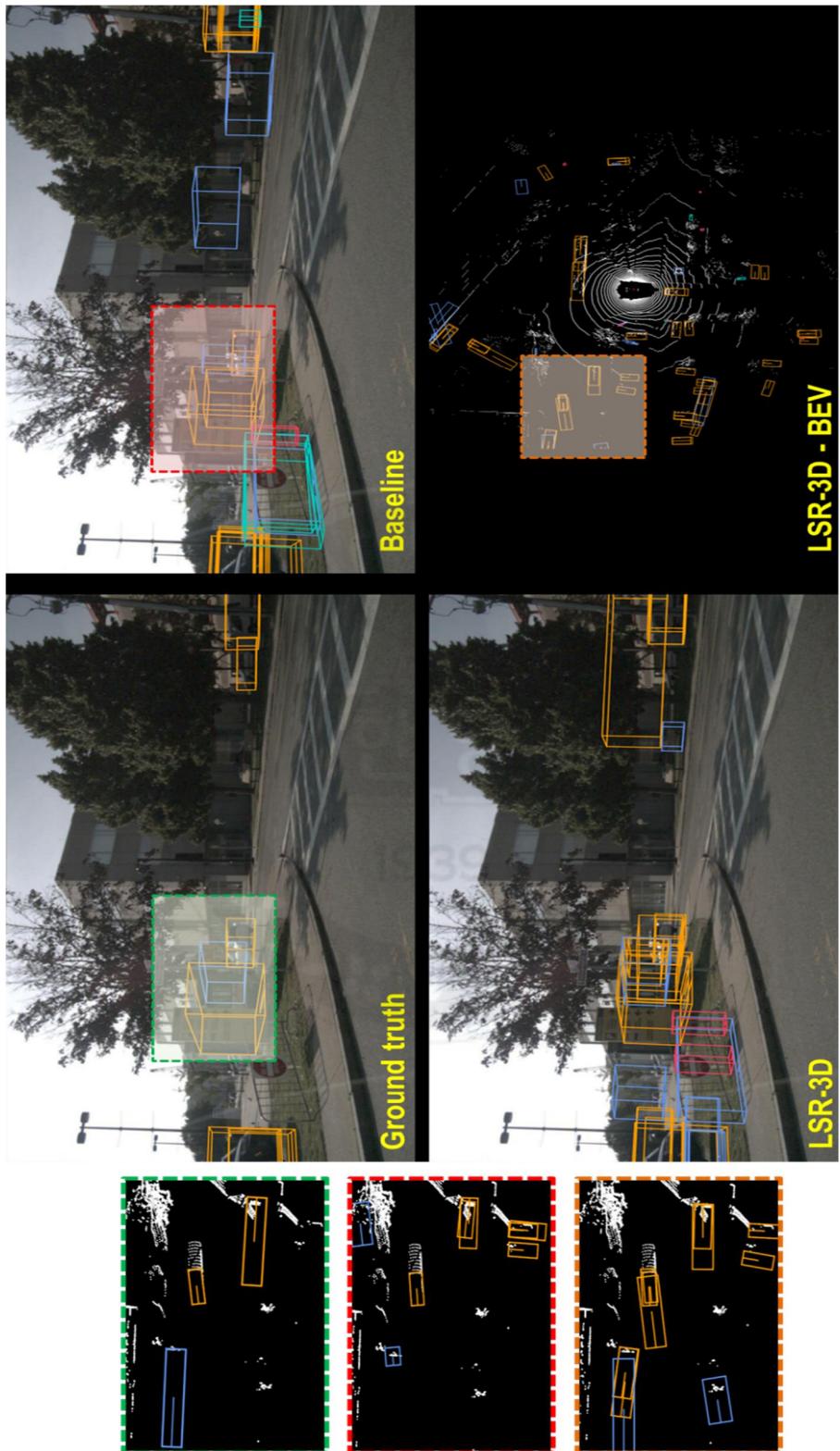


그림 29. Partial view problem 개선에 관한 두 번째 프레임의 검출 결과

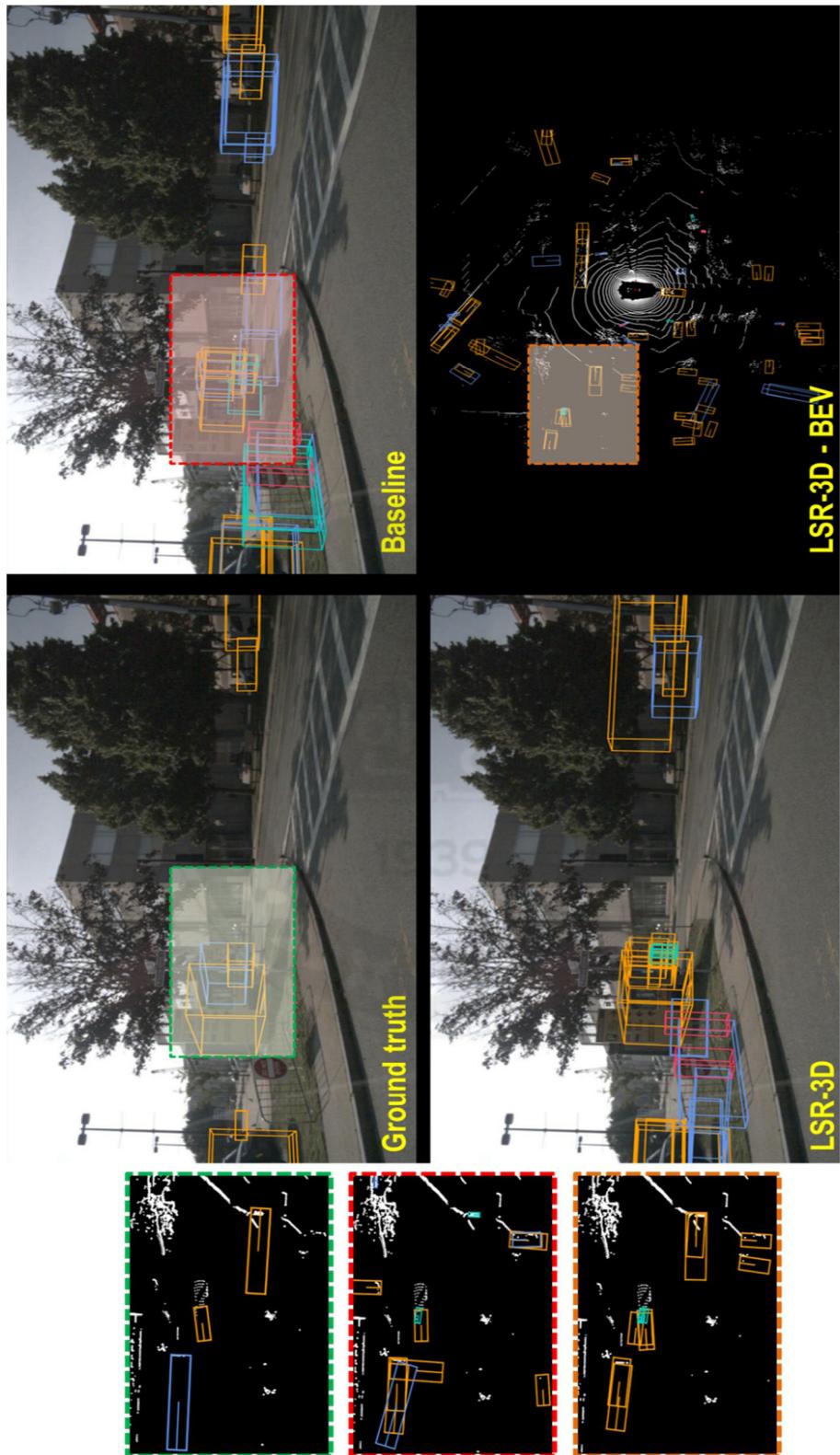


그림 30. Partial view problem 개선에 관한 세 번째 프레임의 검출 결과

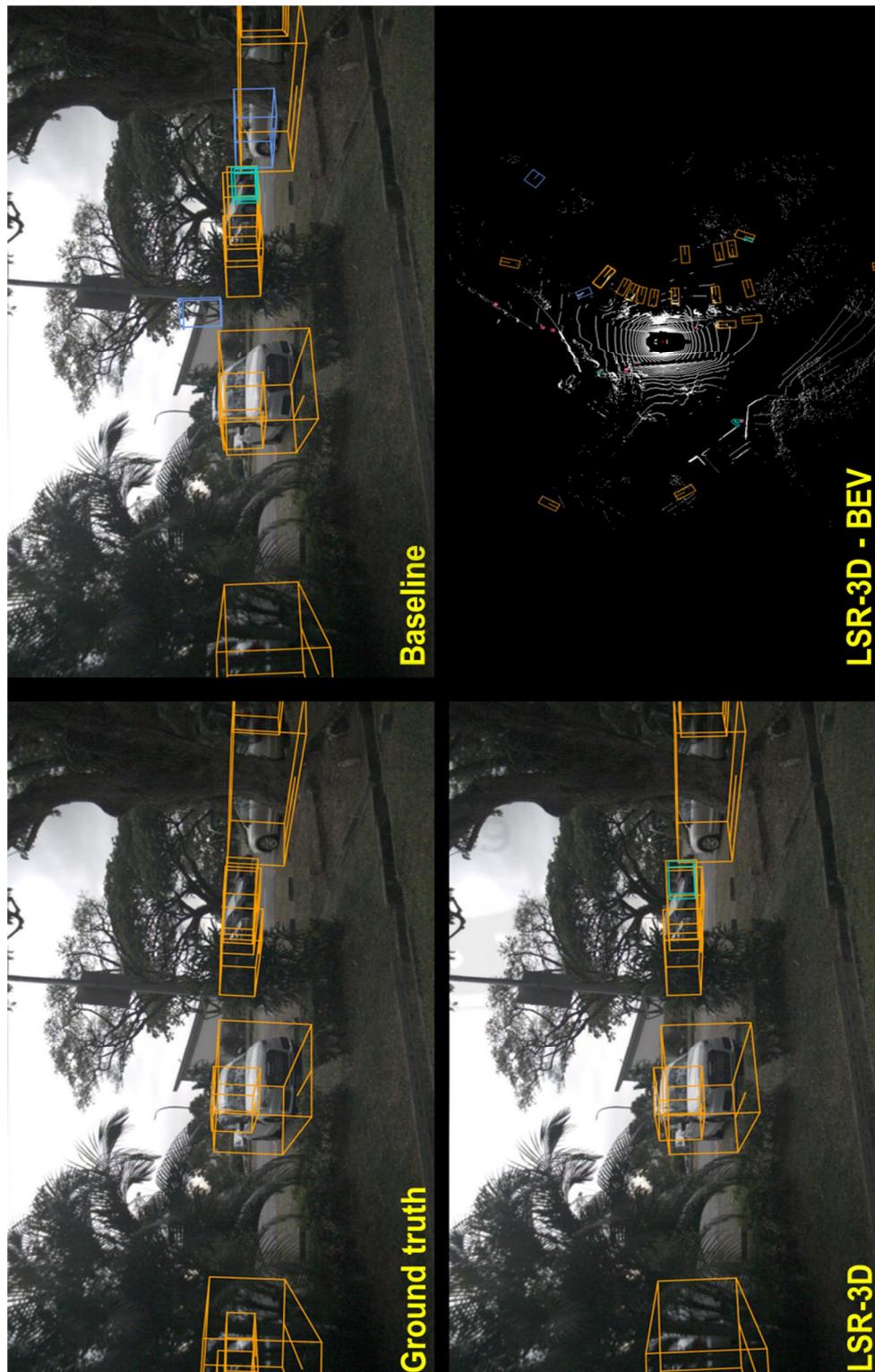


그림 31. 중첩 상황 검출 성능 개선에 관한 첫 번째 프레임의 검출 결과



그림 32. 중첩 상황 검출 성능 개선에 관한 두 번째 프레임의 검출 결과



그림 33. 중첩 상황 검출 성능 개선에 관한 세 번째 프레임의 검출 결과



그림 34. 중첩 상황 검출 성능 개선에 관한 두 번째 예시의 첫 번째 프레임 검출 결과



그림 35. 중첩 상황 검출 성능 개선에 관한 두 번째 예시의 두 번째 프레임 검출 결과



그림 36. 중첩 상황 검출 성능 개선에 관한 두 번째 예시의 세 번째 프레임 검출 결과

제6장 결론

본 논문에서는 포인트 클라우드 시퀀스 기반 3차원 객체 검출을 수행하기 위해 시공간 표현 학습에 관한 딥러닝 알고리즘을 제시하였다. 특히 주어진 포인트 클라우드 시퀀스 데이터를 계층적으로 구성된 단기 시퀀스 관점과 장기 시퀀스 관점으로 구분하고 각 관점의 시공간 특징 표현 학습을 위한 알고리즘을 제안하였다. Short-term aware grid feature encoder 알고리즘은 3차원 공간을 균등하게 분할한 각 격자 공간 영역에 포함된 포인트 클라우드 시퀀스 데이터로부터 시공간 표현을 학습함으로써 종래 방식보다 정보량이 강화된 격자 별 특징 벡터를 도출한다. 한편, 특징 정렬과 특징 결합 두 단계로 구성된 Long-term BEV feature refinement 알고리즘은 각각의 단기 시퀀스 입력으로부터 얻은 특징 표현에 대해 Bird's-eye-view 공간 영역에서 장기 시퀀스에 걸친 시공간 표현을 학습하고 이를 통해 정보량이 강화된 특징 표현을 도출한다. 본 연구에서 제안된 시공간 표현 학습 알고리즘을 종래 기술에 적용한 뒤 nuScenes 데이터셋을 이용해 그 효용성을 검증했으며, 특히 두 가지 알고리즘이 모두 통합된 모델, LSR-3D의 검출 성능이 PointPillars 모델 mAP 성능 대비 5% 이상 향상됨을 확인했다. 또한 제안된 알고리즘을 통해 포인트 클라우드 시퀀스 데이터의 시공간 맥락을 파악함으로써 False positive 검출이 감소하고 중첩이 심한 경우에도 강건한 검출 성능을 보이는 것을 정성적 분석을 통해 확인하였다.



참고 문헌

- [1] Caesar, Holger, et al. "nuscenes: A multimodal dataset for autonomous driving." *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020.
- [2] Lang, Alex H., et al. "Pointpillars: Fast encoders for object detection from point clouds." *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019.
- [3] Feichtenhofer, Christoph, et al. "Slowfast networks for video recognition." *Proceedings of the IEEE/CVF international conference on computer vision*. 2019.
- [4] Feichtenhofer, Christoph. "X3d: Expanding architectures for efficient video recognition." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020.
- [5] Wu, Haiping, et al. "Sequence level semantics aggregation for video object detection." *Proceedings of the IEEE/CVF international conference on computer vision*. 2019.
- [6] Emmerichs, David, Peter Pinggera, and Björn Ommer. "VelocityNet: Motion–Driven Feature Aggregation for 3D Object Detection in Point Cloud Sequences." *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021.
- [7] Sak, Hasim, Andrew W. Senior, and Françoise Beaufays. "Long short–term memory recurrent neural network architectures for large scale acoustic modeling." (2014).
- [8] Cho, Kyunghyun, et al. "Learning phrase representations using RNN encoder–decoder for statistical machine translation." *arXiv preprint arXiv:1406.1078* (2014).
- [9] Shi, Xingjian, et al. "Convolutional LSTM network: A machine learning approach for precipitation nowcasting." *Advances in neural information processing systems* 28 (2015).
- [10] Ballas, Nicolas, et al. "Delving deeper into convolutional networks for learning video representations." *arXiv preprint arXiv:1511.06432* (2015).
- [11] Huang, Rui, et al. "An lstm approach to temporal 3d object detection in lidar point clouds." *European Conference on Computer Vision*. Springer, Cham, 2020.

- [12] Yin, Junbo, et al. "Lidar-based online 3d video object detection with graph-based message passing and spatiotemporal transformer attention." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020.
- [13] Yuan, Zhenxun, et al. "Temporal-channel transformer for 3d lidar-based video object detection for autonomous driving." *IEEE Transactions on Circuits and Systems for Video Technology* 32.4 (2021): 2068–2078.
- [14] Vaswani, Ashish, et al. "Attention is all you need." *Advances in neural information processing systems* 30 (2017).
- [15] Yang, Zetong, et al. "3d-man: 3d multi-frame attention network for object detection." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021.
- [16] Hu, Jie, Li Shen, and Gang Sun. "Squeeze-and-excitation networks." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.
- [17] Deng, Jiajun, et al. "Voxel r-cnn: Towards high performance voxel-based 3d object detection." *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. No. 2. 2021.
- [18] Shi, Shaoshuai, et al. "Pv-rcnn: Point-voxel feature set abstraction for 3d object detection." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020.
- [19] Bhattacharyya, Prarthana, Chengjie Huang, and Krzysztof Czarnecki. "Sa-det3d: Self-attention based context-aware 3d object detection." *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021.
- [20] Ioffe, Sergey, and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift." *International conference on machine learning*. PMLR, 2015.
- [21] Zhu, Xizhou, et al. "Deformable convnets v2: More deformable, better results." *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019.
- [22] Lin, Tsung-Yi, et al. "Feature pyramid networks for object detection." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.
- [23] Hyeok, Yoo Jin, Kum Dongsuk, and Choi Jun Won. "Scarfnet: Multi-scale features with deeply fused and redistributed semantics for enhanced object detection." *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2021.

- [24] Wang, Xiaolong, et al. "Non-local neural networks." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.
- [25] Geiger, Andreas, et al. "Vision meets robotics: The kitti dataset." *The International Journal of Robotics Research* 32.11 (2013): 1231–1237.
- [26] Yin, Tianwei, Xingyi Zhou, and Philipp Krahenbuhl. "Center-based 3d object detection and tracking." *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021.
- [27] Yan, Yan, Yuxing Mao, and Bo Li. "Second: Sparsely embedded convolutional detection." *Sensors* 18.10 (2018): 3337.



ABSTRACT

Learning Spatiotemporal Representation of Point Cloud Sequences for 3D Object Detection

Lee, Jun Hyung

Dept. of Future Mobility

Graduate School of

Hanyang University

LiDAR sensor, which creates a point cloud based on laser pulses reflected on the surface of an object, provides accurate 3D spatial information, so it serves as a key sensor for perception tasks in various mobility industries, including autonomous driving and robotics. Recently, as deep learning technology has rapidly advanced, research for deriving feature representations from point clouds and exploiting them for 3D object detection is being actively conducted. On the other hand, algorithms that utilize spatiotemporal information included in sequential data have been proven effective in various image recognition fields, including action recognition and object tracking. The LiDAR sensor also creates a point cloud sequence in real time through continuous scanning. However, conventional studies have focused on a 3D object detection algorithm based on a single scan result. Even if sequence data are used, they are only to use a higher-density point cloud obtained through a simple merging process.

Therefore, in this study, by utilizing the spatiotemporal information included in the point cloud sequence, we tried to improve the following two problems that have been analyzed as limiting factors in improving the performance of lidar sensor-based object detection. First, we tried to improve the problem of point data sparsity according to the LiDAR sensor resolution. Second, we tried to improve the partial acquisition problem according to the positional relationship between the sensor and the object. To this end, we viewed point cloud sequences as a hierarchical structure divided into short-term and long-term sequences and then proposed deep learning algorithms for each perspective that can learn and utilize spatiotemporal feature representations based on the hierarchical perspective. In addition, we designed the point cloud sequence-based 3D object detection pipeline so that each proposed algorithm could be applied to the conventional detectors.

Supervised learning and inference experiments were conducted on the proposed algorithms using the nuScenes dataset, which is a representative autonomous driving dataset, and quantitative and qualitative analyzes were performed based on the experimental results. The proposed method in this paper shows a performance improvement of 5.54% and 3.07% in mAP and NDS performance, respectively, compared to PointPillars, a baseline model, for the nuScenes validation dataset, thereby demonstrating the effectiveness of learning the spatiotemporal feature representation from point cloud sequences. In particular, the high performance gains were obtained in motorcycle, bicycle, and pedestrian classes with relatively small aspect ratios in bird's-eye-view and fewer acquired points compared to vehicle-related classes through spatiotemporal representation learning based on hierarchical views. The qualitative analysis confirmed that the false positive rate decreased, and our proposed method performed robust detection even when occlusion occurred between objects.



연구 윤리 서약서

본인은 한양대학교 대학원생으로서 이 학위논문 작성 과정에서 다음과 같이 연구 윤리의 기본 원칙을 준수하였음을 서약합니다.

첫째, 지도교수의 지도를 받아 정직하고 엄정한 연구를 수행하여 학위논문을 작성한다.

둘째, 논문 작성시 위조, 변조, 표절 등 학문적 진실성을 훼손하는 어떤 연구 부정행위도 하지 않는다.

셋째, 논문 작성시 논문유사도 검증시스템 "카피킬러"등을 거쳐야 한다.

2022년12월19일

학위명 : 석사

학과 : 미래모빌리티학과

지도교수 : 최준원

성명 : 이준형



한 양 대 학 교 대 학 원 장 귀 하

Declaration of Ethical Conduct in Research

I, as a graduate student of Hanyang University, hereby declare that I have abided by the following Code of Research Ethics while writing this dissertation thesis, during my degree program.

"First, I have strived to be honest in my conduct, to produce valid and reliable research conforming with the guidance of my thesis supervisor, and I affirm that my thesis contains honest, fair and reasonable conclusions based on my own careful research under the guidance of my thesis supervisor.

Second, I have not committed any acts that may discredit or damage the credibility of my research. These include, but are not limited to : falsification, distortion of research findings or plagiarism.

Third, I need to go through with Copykiller Program(Internet-based Plagiarism-prevention service) before submitting a thesis."

DECEMBER 19, 2022

Degree : Master

Department : DEPARTMENT OF FUTURE MOBILITY

Thesis Supervisor : Jun-Won Choi

Name : LEE JUNHYUNG

(Signature)

