

기술 백서: Real-time Motion Planning Framework with Learned Committed Trajectory Distribution

[Paper Title / Core Technology]

Real-time Motion Planning Framework for Autonomous Vehicles with Learned Committed Trajectory Distribution (Minsoo Kim et al., IROS 2023)

1. 설계 철학 및 문제 정의 (Architectural Philosophy)

기존 기술의 임계점 (Legacy Bottleneck):

- 비효율적 전역 예측 (Inefficient Global Bias): 기존의 학습 기반 모션 플래닝 (Deep Learning-based Motion Planning)은 전체 최적 경로 (Whole Optimal Path)를 한 번에 예측하여 샘플링 편향 (Biasing)을 유도했다.
- 실시간성 괴리 (Real-time Mismatch): 그러나 Anytime Framework (실행과 계획을 동시에 수행)에서는 차량이 당장 실행할 경로 (Committed Trajectory)가 가장 중요하다. 먼 미래의 경로까지 예측하는 것은 계산 자원을 낭비할 뿐만 아니라, 동적 환경이나 좁은 공간에서 부정확한 예측을 초래하여 오히려 수렴 속도를 저하시킨다.

패러다임 전환 (Paradigm Shift):

- 순차적 집중 (Sequential Focus): 본 논문은 전체 경로가 아닌, "차량이 당장 실행해야 할 다음 구간 (Committed Trajectory)"만을 단계별로 예측하는 Learned Anytime Predictor를 제안한다.
- Local-Optimization First: 전역적인 최적해를 한 번에 찾으려 하기보다, 실행 루프 (Control Loop, approx 20ms) 내에서 가장 확실한 다음 스텝을 확률 분포로 예측하고 이를 기반으로 RRT*를 확장하여 실시간성과 정확성을 동시에 확보한다.

개념 시각화 (Conceptual Analogy):

> [Analogy] 짙은 안개 속에서 운전할 때, 집까지 가는 전체 경로를 모두 머릿속에 그리고 운전하는 것(기존 방식)은 불가능하고 위험하다. 대신, 헤드라이트가 비추는 "당장 앞 10미터" (Committed Trajectory)를 정확하게 파악하고 그 구간을 최적으로 통과하는 데 집중하는 것(제안 방식)이 훨씬 빠르고 안전하다.

2. 수학적 원리 및 분류 (Mathematical Formalism)

시스템 분류 (System Taxonomy):

- 알고리즘 계열: Sampling-based Motion Planning (RRT*)

- 프레임워크: Anytime Framework (Plan-with-Execution)
- 학습 모델: CNN-based Encoder-Decoder (Lightweight)

핵심 수식 및 상세 해설 (Core Formulation & Breakdown):

1. Optimization Objective (Committed Trajectory Sequence):

$$\pi_{c,k}^* = \operatorname{argmin}_{\pi_{c,k}} \sum_{q_s} q_g \operatorname{cost}(\pi_{c,k})$$

- Variable Definition:
- $\pi_{c,k}$: k번째 Committed Trajectory (트리에서 확정된 경로 조각).
- q_s, q_g : 시작 및 목표 지점.
- Physical Meaning:
- 전체 경로 비용을 최소화하는 문제는 결국 "일련의 최적 Committed Trajectory 조각들을 순차적으로 찾아내는 문제"로 귀결된다. 이전 조각($\pi_{c,k-1}$)이 결정되어야 다음 조각($\pi_{c,k}$)을 최적화할 수 있는 순차적 의존성(Sequential Dependency)을 강조한다.

2. Network Loss Function (Classification + Regression):

$$\text{Loss} = \sum_{g \in G} \left(f_{ce}(g) L_{ce}(g) + f_{mse}(g) L_{mse}(g) \right)$$

- Variable Definition:
- L_{ce} : Grid-wise Cross-Entropy Loss (해당 그리드가 경로에 포함될 확률).
- L_{mse} : Mean Squared Error (해당 그리드에서의 차량 방향 $\cos\theta, \sin\theta$).
- $f(g)$: Weighting Function (경로 위 그리드에 가중치 $w=20$ 부여).
- Physical Meaning:
- 위치 예측(Classification)과 방향 예측(Regression)을 동시에 수행한다. 특히, 대부분의 공간이 빈 공간(Empty)인 OGM(Occupancy Grid Map) 특성을 고려하여, 실제 경로가 존재하는 그리드에 높은 가중치($f(g)$)를 두어 학습 불균형(Class Imbalance)을 해소하고 수렴을 가속화한다.

3. 실행 파이프라인 및 데이터 흐름 (Execution Pipeline)

입력 명세 (Input Context):

- Local Map: 5-Channel Grid Map (Occupied, Unknown, Root, Goal, Current π_c).
- State: Vehicle Pose (x, y, θ).

순전파 로직 (Forward Propagation with Anytime Planning):

1. Initial Planning Phase:

- Prediction: Start Node(q_{start})와 Goal(q_g)을 입력받아 첫 번째 구간 분포 $\mu_{c,1}$ 예측.
- Sampling: N_{bias} (Learned Distribution) + N_{uni} (Uniform) 샘플링.

- Tree Expansion: RRT*를 통해 트리 확장 및 초기 경로 생성.

2. Iterative Planning Phase (Loop):

- Commit: 일정 시간(t_{init}) 후, 현재 트리의 Best Path 중 첫 번째 엣지를 $\pi_{c,1}$ 로 확정(Commit)하고 차량 이동 시작.
- Next Prediction: 차량이 $\pi_{c,1}$ 을 따라가는 동안, 네트워크는 다음 구간($\pi_{c,2}$)의 분포 $\mu_{c,2}$ 를 미리 예측.
- Refinement: 예측된 $\mu_{c,2}$ 를 중심으로 샘플링을 집중하여 다음 경로 최적화.
- Repeat: 목표 도달 시까지 반복.

Data Augmentation Strategy:

- 하나의 긴 최적 경로(Whole Path)를 여러 개의 Committed Trajectory 조각으로 쪼개어 학습 데이터를 증강함.
- Effect: 동일한 맵/목표에서도 다양한 시작점(Root)과 중간 경로 상황을 학습하게 되어, 네트워크가 다양한 부분 경로 상황에 강건(Robust)해짐.

4. 학습 메커니즘 및 최적화 (Optimization Dynamics)

학습 전략 (Training Dynamics):

- Lightweight Architecture:
- 기존 모델 대비 채널 수를 1/4로 줄인 경량 CNN 사용.
- Reason: 통신 및 제어 루프(약 20ms) 내에서 추론과 샘플링을 모두 끝내야 하므로, 정확도보다 실시간 추론 속도가 최우선.
- Result: 추론 + 100개 샘플 추출에 20ms 미만 소요.

알고리즘 비교 (Performance gains):

- vs Hierarchical (Plan-then-Execute): 멈춰서 계획할 필요가 없어 주행 시간 대폭 단축.
- vs Whole-Path Prediction:
- 복잡한 교차 구간(Switching back and forth)에서 전체 경로 예측은 실패할 확률이 높음.
- 본 방식은 구간별로 쪼개서 예측하므로 복잡한 조작이 필요한 주차 시나리오(S6~S10)에서 시간을 최대 30초 단축.

5. 구현 상세 및 제약 사항 (Details & Constraints)

테스트 환경 (Environment Spec):

- Simulator: CARLA (Unstructured Parking Scenarios).

- Controller: Kanayama Controller (20ms Loop).
- Steer Function: Hybrid Curvature (Non-holonomic 제약 고려).

시스템 한계 (Limitations):

- Dynamic Obstacles: 정적 장애물 중심의 실험. 움직이는 장애물에 대한 반응성은 추가 검증 필요.
- Map Dependency: 사전에 정의된 Grid Map 해상도(0.2m/px)에 의존적. 해상도가 바뀌면 재학습 필요 가능성.

6. 산업 적용 전략 (Industrial Application)

Target Industry:

- Autonomous Valet Parking (AVP): 좁고 복잡한 주차장에서 정밀하고 신속한 경로 생성이 필수적인 분야.
- Logistics Robots (AMR): 물류센터 내에서 예상치 못한 장애물을 피해 최단 거리로 이동해야 하는 로봇.

Business Value:

- Time Saving: 주차 시간 38% 단축 서비스 회전율 증가.
- Smoothness: 정지 후 재계획(Stop-and-Go)이 아닌 연속적인 주행으로 승객 경험(Ride Comfort) 향상.