

도로 네트워크 분할 기반의 효율적인 대규모 정밀지도 원호 매개변수화 기법

전진환¹⁾·최세범^{*1)}

한국과학기술원 기계공학과¹⁾

Road Network Decomposition based Efficient Arc Spline Approximation of Large Sized HD Maps

Jinhwan Jeon¹⁾ · Seibum B. Choi^{*1)}

¹⁾ Department of Mechanical Engineering, Korea Advanced Institute of Science and Technology,
291 Daehak-ro, Yuseong-gu, Daejeon 34141, Korea

Abstract : Advancements in autonomous driving and ADAS have increased the demand for scalable HD lane-level maps that accurately capture geometric properties. Traditional polyline-based formats struggle with continuous curvature and heading representation, leading to the development of curve-based methods as an improvement. However, existing curve-based approaches are often limited to simplified road networks. This study presents an arc-spline-based lane approximation that efficiently models large-scale road networks while preserving geometric continuity. By introducing a novel road network decomposition and localized optimization method, our approach enhances computational efficiency without requiring full map-scale optimization. Experiments on the nuScenes dataset demonstrate a 4.3 cm average approximation error, maintaining lane topology and connectivity while significantly reducing storage requirements. Resulting figures can be viewed at our github repository (<https://github.com/wjswlsghks98/Efficient-Arc-Spline-Approximation>).

Key words : Autonomous Driving(자율주행), High Definition Maps (정밀지도), Road Network Decomposition(도로 네트워크 분할), Arc Spline Approximation (원호 매개변수화)

Nomenclature

A : arc node vector $\in \mathbb{R}^{2 \times 1}$
P : data point with noise $\in \mathbb{R}^{2 \times 1}$
l : linestring (sequence of 2D points)
A, B : linestring clusters type A and B
k : signed distance arc parameter $\in \mathbb{R}$
 Σ : measurement covariance
 \mathcal{L} : cost function
r : cost function residual vector
 $G(V, E)$: road network graph (vertices, edges)

Subscripts

AC : Anchor model cost function
ME : Measurement model cost function

1. 서론

지속적인 자율주행 기술의 발전에 따라, 정밀지도(HD Map)의 활용 범위가 특정 구역이나 제한된 임무 수행에서 벗어나, 넓은 영역에서의 고차

원 임무 수행¹⁾을 지원하는 방향으로 확대되고 있다. 이에 따라 대규모 정밀지도의 제작이 활발히 이루어지고 있으며, 현재 많이 활용되는 지도 포맷들²⁻⁵⁾은 차선 데이터를 점으로 저장하고 이를 직선으로 연결하는 polyline 형식을 채택하고 있다. 이러한 방식은 데이터 구조가 단순하여 다루기 용이하지만, 도로의 곡률, 헤딩 각과 같은 연속적인 기하학적 특성이 반영되지 않는 한계가 있다. 또한, 높은 곡률을 가지는 도로에서는 차선을 정확히 표현하기 위해 지나치게 많은 점이 필요하여 저장 효율이 저하되는 문제도 존재한다.

이러한 한계를 극복하기 위해 곡선 기반의 차선 근사 방식들⁶⁻¹⁴⁾이 제안되었으며, 이는 polyline 형식보다 연속적인 기하학적 정보를 보존하는 장점을 가진다. 그러나 기존 연구들은 단순하게 연결된 도로망만을 다루거나, 교차로가 포함된 경우에도 넓지 않은 지역을 대상으로만 매개변수화를 수행하는 범용성의 한계를 가지고 있다. 넓은 영역을 다루면

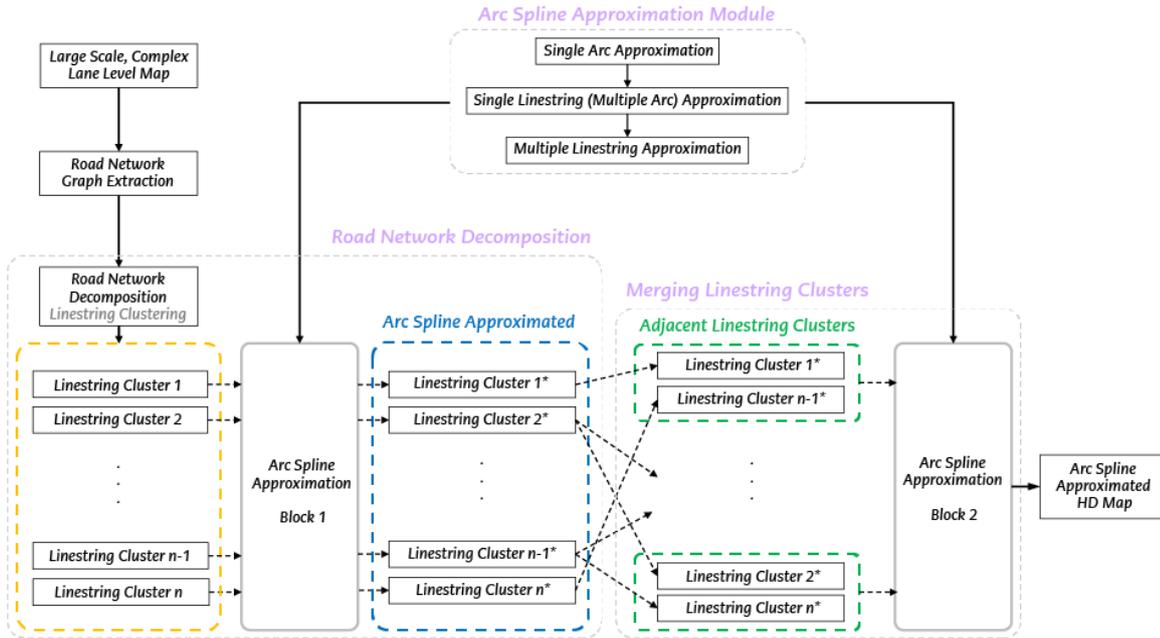


Fig. 1 도로 네트워크 분할 및 효율적인 원호 매개변수화 기법에 대한 전체 프레임워크. 제2장에서는 상단의 원호 매개변수화 모듈을 다루고, 제3장에서는 그림 왼쪽 하단과 오른쪽 하단과 같이 도로 네트워크 분할과 분할된 지도들을 각각 매개변수화한 후 융합하는 내용을 소개한다.

서 복잡하게 연결된 도로망을 곡선 기반으로 매개변수화하기 위해서는 두 가지 핵심 문제를 해결해야 한다. 첫째, 복잡하게 연결된 차선들이 최소한 G1 연속성(기울기 연속성)을 만족해야 한다. 둘째, 넓은 영역을 대상으로 매개변수화를 수행해야 한다는 점이다.

본 연구에서는 첫 번째 문제를 해결하기 위해 기존의 원호 매개변수화 연구¹⁴⁾를 확장하여 복잡한 도로망에서도 G1 연속성을 보장하는 방식을 제안한다. 두 번째 문제와 관련해서는 대규모 지도를 작은 단위로 분할한 후, 각 소규모 지도에 대해 매개변수화를 수행하고, 이를 효율적으로 융합하는 방법을 다룬다.

제2장에서는 복잡한 도로망에서 G1 연속성을 확보하는 과정, 제3장에서는 대규모 지도에서의 매개변수화 및 융합 기법을 상세히 설명하며, 제4장에서는 제안한 알고리즘을 실제 데이터셋에 적용한 결과와 분석을 제시한다. 전체 연구에 대한 흐름도는 Fig. 1과 같다.

2. 차선 원호 매개변수화 모듈

본 장에서는 복잡하게 연결된 차선들이 G1 연속성을 유지하도록 기존 연구의 원호 모델을 확장하는 방법을 다룬다. 이를 본격적으로 설명하기 전

에, G1 연속성이 필요한 이유와 본 연구에서 원호 기반 매개변수화를 선택한 이유를 간략히 정리한다.

G1 연속성은 자율주행과 ADAS에서 경로 생성과 제어에 필수적인 요소로, 이를 충족하지 않으면 차선 정보의 불연속성으로 인해 경로 추종 제어 성능이 저하될 수 있다. 또한 본 연구에서는 원호를 선택했는데, 원호는 회전과 병진 이동에 대해 기하학적 특성이 변하지 않으며, 특정 점과의 거리를 대수적으로 계산할 수 있다는 장점이 있다. 이를 통해 차량 좌표계로 변환이 용이하고, 차선과의 거리를 실시간으로 정확하게 측정할 수 있어 효과적인 차선 정보의 활용이 가능하다.

2.1 단일 원호 매개변수화

원호를 표현하는 방법은 매우 다양하지만, 그중에서 본 연구에서는 유리 베지에 (Rational Bezier)형식을 활용하여 하나의 원호를 표현하고, 이전 연구에서는 위의 그림에서 A_1, A_2, C 총 3개의 2D 점, 즉 6개의 파라미터를 하나의 원호에 할당했다면 본 연구에서는 Fig. 2의 (a)에서 볼 수 있듯이 2개의 2D 점 (A_1, A_2)과 부호가 있는 거리상수 $k \in \mathbb{R}$ 로 하나의 원호를 표현한다. 이는 곡선의 control point인 C를 변수로 둘 필요 없이 아래와 식을 유도할 수 있기 때문이다.

$$\mathbf{v} = \frac{1}{\|\mathbf{A}_1 - \mathbf{A}_2\|_2} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \cdot (\mathbf{A}_1 - \mathbf{A}_2)$$

$$\mathbf{C} = \frac{1}{2}(\mathbf{A}_1 + \mathbf{A}_2) + k\mathbf{v}$$

(1)

\mathbf{A}_1 , \mathbf{A}_2 , 그리고 k 가 최적화 변수인 최적화 문제를 구성할 수 있는데, 이 형태는 흔히 Nonlinear Least Squares(NLS)로 알려져 있다. 본 연구에서는 MATLAB의 'lsqnonlin.m' 솔버를 활용하여 최적의 원호 파라미터들을 얻었다.

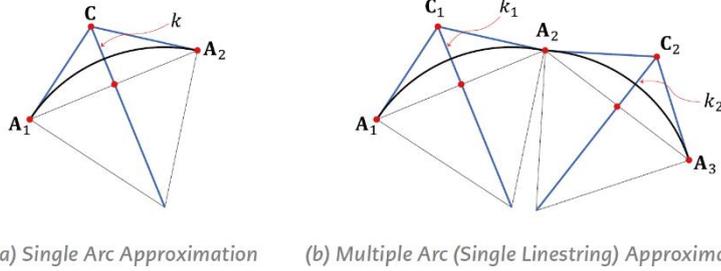


Fig. 2 변형된 유리 베지에 형식 기반의 원호 모델링 (a) 단일 원호, (b) 다중 원호 (단일 Linestring)

위와 같이 원호를 모델링 한 후, 노이즈가 포함된 점들을 정확하게 근사하는 원호를 구하기 위해서 최적화를 위한 비용함수를 구성하였다.

2.1.1 Anchor 모델

Anchor 모델은 표현 그대로, 원호의 양 끝점 \mathbf{A}_1 과 \mathbf{A}_2 를 점군 데이터의 각각 처음과 끝점에 정확하게 고정(앵커)하기 위한 비용함수 모델로, 해당 점까지의 거리를 비용함수로 설정하여 거리가 멀어질수록 비용이 커지도록 구성하였고, 식은 아래와 같다.

$$\mathcal{L}_{AC} = \|\mathbf{P}_1 - \mathbf{A}_1\|_{\Sigma_{AC}}^2 + \|\mathbf{P}_n - \mathbf{A}_2\|_{\Sigma_{AC}}^2 \quad (2)$$

이때, $\|\cdot\|_R^2$ 의 표현은 Squared Mahalanobis Distance로, 아래와 같이 표현된다.

$$\|\theta - \mu\|_R^2 \triangleq (\theta - \mu)^T \Sigma^{-1} (\theta - \mu) \quad (3)$$

Σ_{AC} 값은 Anchor 모델의 공분산으로, tunable parameter이다. 아래에 나올 Measurement 모델과의 비용함수 비율을 바꿀 때 Σ_{AC} 의 값을 조정한다.

2.1.2 Measurement 모델

Measurement 모델은 \mathbf{A}_1 , \mathbf{A}_2 , 그리고 k 을 통해 구성된 원호가 주어진 점군 데이터를 정확하게 근사할 수 있도록, 근사 오차를 비용함수로 설정한다. 해당 함수에 대한 구체적인 유도 과정은 이전 연구¹⁴⁾와 거의 동일하기 때문에 생략하였다. 수식은 아래와 같다.

$$\mathcal{L}_{ME} = \sum \|\mathbf{r}_{ME}^i(\mathbf{A}_1, \mathbf{A}_2, k, \mathbf{P}_i)\|_{\Sigma_{ME}^i}^2 \quad (4)$$

2.1.3 최적화 방법

위의 두 모델로부터 얻은 비용함수를 토대로,

2.2 단일 Linestring(다중 원호) 매개변수화

일반적으로 하나의 점 시퀀스(Linestring)에 포함된 점군 데이터는 단 하나의 원호만으로 정확하게 표현하기 어렵다. 따라서, Fig. 2(b)와 같이 여러 개의 원호를 활용하여 근사해야 한다. 이를 효율적으로 모델링하기 위해 인접한 원호들은 노드를 공유하도록 설정하며, 예를 들어 Fig. 2(b)에서는 두 개의 원호를 총 3개의 노드와 2개의 거리 상수로 표현할 수 있다. 그러나 원호가 하나일 때와 달리, 여러 원호를 사용할 경우 G1 연속성(기울기 연속성)을 유지하기 위한 추가적인 제약 조건이 필요하다. 즉, 두 원호가 만나는 점 \mathbf{A}_2 에서 기울기가 연속이 되도록 해야 한다. 일반적으로, 원호가 m 개 일 때 이를 만족하는 제약 조건은 다음과 같다.

$$r_{Eq}(i) = \frac{(\mathbf{A}_{i+1} - \mathbf{C}_i)^T (\mathbf{C}_{i+1} - \mathbf{A}_{i+1})}{\|\mathbf{A}_{i+1} - \mathbf{C}_i\| \|\mathbf{C}_{i+1} - \mathbf{A}_{i+1}\|} - 1, \quad i = 1:m \quad (5)$$

단일 차선(Linestring)에 대한 전체 비용함수는 앞서 소개했던 단일 원호에 대한 두 가지 모델을 모든 원호들에 대해 적용 후 더하고, 인접한 원호들 간에 식(5)를 계산하여 제약 조건을 추가하면 된다. 단일 원호와 마찬가지로 MATLAB의 솔버를 통해서 최적의 파라미터들을 구한다.

2.3 다중 Linestring 매개변수화

앞서 하나의 차선(Linestring)을 여러 개의 원호로 근사하는 방법을 다루었는데, 이제 이를 확장하여 복잡하게 연결된 도로 네트워크를 매개변수화하면서 전역적인 G1 연속성을 보장하는 방법을 소개한다.

각 차선은 2.2절에서 제시한 방법을 통해 원호로 매개변수화할 수 있지만, 인접한 차선들 사이에서는 G1 연속성이 자동으로 보장되지 않기 때문에 추

가적인 제약 조건이 필요하다. 이를 위해 전체 비용함수를 구성할 때, 각 차선에 대해 2.2절의 비용함수를 적용하는 것뿐만 아니라, 인접 차선의 원호들이 원활하게 연결될 수 있도록 식 (5)를 활용한 추가적인 제약 조건을 함께 고려해야 한다. 총 s 개의 Linestring이 있을 때 전체 비용함수는

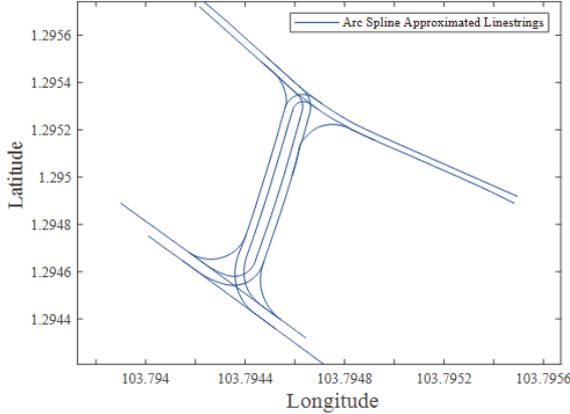


Fig. 3 복잡하게 연결되어 있는 여러 Linestring을 원호로 매개변수화한 예시

다음 수식과 같이 표현할 수 있다.

$$\begin{aligned} \min_{\mathbf{A}_i, \mathbf{k}} \mathcal{L} &= \sum_{l=1}^s \left[\left\| \mathbf{P}_1^{(l)} - \mathbf{A}_1^{(l)} \right\|_{\Sigma_{AC_1}}^2 + \left\| \mathbf{P}_n^{(l)} - \mathbf{A}_{m+1}^{(l)} \right\|_{\Sigma_{AC_1}}^2 \right. \\ &+ \sum_{i=2}^{m^{(l)}} \left\| \mathbf{P}_{\text{Idx}(i)}^{(l)} - \mathbf{A}_i^{(l)} \right\|_{\Sigma_{AC_2}}^2 \\ &\left. + \sum_{i=1}^{m^{(l)}} \sum_{j=\text{Idx}(i)}^{\text{Idx}(i+1)} \left\| \mathbf{r}_{ME}^i \left(\mathbf{A}_i^{(l)}, \mathbf{A}_{i+1}^{(l)}, \mathbf{k}_i^{(l)}, \mathbf{P}_j^{(l)} \right) \right\|_{\Sigma_{ME}}^2 \right] \\ \text{s.t. } \mathbf{r}_{Eq}^{(1)}, \dots, \mathbf{r}_{Eq}^{(s)} &= \mathbf{0}, \mathbf{r}_{Eq2}^{(E)} = \mathbf{0} \end{aligned} \quad (6)$$

해당 식에서 $\mathbf{r}_{Eq2}^{(E)}$ 는 각 차선 간의 연결정보(Edge, E)를 이용해서 식(5)를 인접 차선들 사이의 원호들에 적용한 것이다. 식(6)을 기반으로 복잡하게 연결되어 있는 여러 Linestring들을 원호로 매개변수화한 결과는 Fig. 3과 같다.

2.4 한계점

2.3절에서 제안한 다중 Linestring 매개변수화 방법은 복잡하게 연결된 차선들에 대해 전역적인 G1 연속성을 유지하면서 원호로 근사할 수 있지만, 적용 범위가 넓어질수록 한계가 뚜렷해진다.

식 (6)의 구조를 보면, 대규모 지도를 매개변수화할 경우 최적화해야 할 변수와 비선형 제약 조건이 급격히 증가하여 연산 비용이 크게 늘어난다. 특히, 매개변수화된 지도를 갱신하거나 확장할 때 마다 전체 지도를 다시 최적화해야 하므로 비효율

적이다.

이를 해결하기 위해, 다음 장에서는 지도를 더 작은 단위로 나누어 개별적으로 최적화한 후, 이를 효율적으로 융합하는 방법을 제안한다.

3. 도로 네트워크 분할 기법 및 효율적인 클러스터 융합 방법

본 장에서는 복잡하게 연결된 대규모 도로 네트워크를 여러 개의 원호로 매개변수화하는 과정에서 효율성을 높이는 접근 방법을 소개한다.

3.1 도로 네트워크의 분할

앞서 2.4절에서 언급한 것처럼, 지도 전체를 한번에 매개변수화하는 것은 비효율적이므로, 지도를 더 작은 단위로 나눈 후 개별적으로 매개변수화하고, 이를 융합하는 분할정복(Divide and Conquer) 접근법이 필요하다.

Fig. 3과 같이 차선들이 복잡하게 연결된 경우, 차선을 Vertex(V)로, 차선 간 연결이 있는 경우 Edge(E)를 1로 설정하여 그래프 $G(V, E)$ 를 생성할 수 있다. 이 그래프는 차선 간의 연결 관계를 명확하게 나타내므로, 앞서 제시한 분할정복 방식 적용 시 그래프 G 를 적절히 분할하는 것이 효과적이다.

3.1.1 기존 그래프 분할 기법

일반적으로 그래프 $G(V, E)$ 를 분할하는 방법은 그래프 이론 및 네트워크 이론 분야에서 활발히 연구되어 왔으며, 그중 본 연구에 적용할 수 있는 대표적인 기법으로 Graph Component와 Louvain¹⁵⁾방법이 있다.

먼저, Graph Component는 엄밀히 말해 그래프 분할 기법은 아니지만, 주어진 그래프에서 서로 연결된 요소들을 하나의 클러스터로 묶는 방식이기 때문에 차선 지도에서 고립된 구역을 효과적으로 식별하고 매개변수화를 수행하는 데 유용할 것으로 판단된다. 다만, 복잡하게 연결된 대규모 지도에서는 모든 차선이 하나의 연결 요소로 인식되어 결국 전체 지도에 대한 최적화를 수행해야 할 가능성이 높아지는 한계점이 있다.

Louvain 방법은 주어진 그래프에서 Vertex 간 연결 밀도가 높은 구역(Community)을 감지하여 이를 기준으로 그래프를 분할하는 기법이다. 이 방법은 복잡하게 연결된 교차로 등의 영역을 효과적으로 분리할 수 있지만, 이후 클러스터들을 융합하는 과정에서 문제가 발생한다. 분할된 소규모 Linestring 클러스터들을 2.3절에 따라 개별적으로 원호 매개변수화한 후, 이들을 하나의

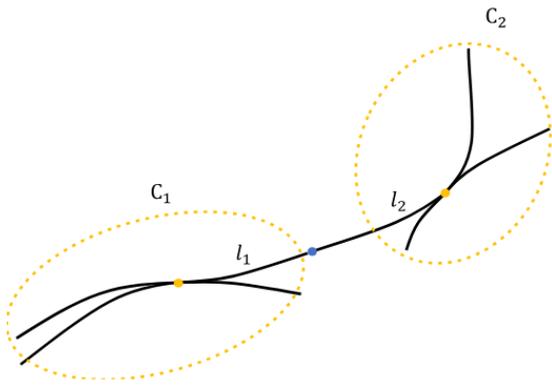


Fig. 4 인접한 Linestring 클러스터를 효율적으로 융합하기 위한 조건. 인접한 클러스터 C1과 C2가 있을 때, 그림과 같이 l_1 과 l_2 가 단순히 직렬로 연결되어 있으면 C1과 C2에 속한 전체 Linestring들에 대한 최적화가 아닌, l_1 과 l_2 에 대한 소규모 최적화 문제만 풀어도 충분하다.

지도 형태로 결합해야 하는데, 이때 인접한 클러스터들 간 G1 연속성을 유지하기 위한 추가적인 최적화가 필요하다. 인접한 클러스터를 융합하는 과정에서 두 클러스터에 포함된 모든 Linestring을 최적화 대상으로 설정해야 하는 경우, 최종적으로 전체 지도를 다시 최적화해야 하는 상황이 발생할 수 있다. Louvain 방법뿐만 아니라, 다른 일반적인 그래프 분할 기법을 적용해도 동일하게 이와 같은 문제가 초래되며, 결국 지도 전체를 최적화하면서 발생하는 병목 현상을 해결하지 못한다.

따라서, 이러한 문제를 극복하기 위해 새로운 그래프 분할 방법을 제안하며, 이를 통해 최적

화 과정에서 발생하는 비효율성을 해결하는 방법을 다음 절에서 자세히 설명한다.

3.1.2 도로 네트워크 분할 규칙

도로 네트워크 그래프를 분할한 후, Fig. 4와 같이 Linestring 클러스터가 생성되었다면, 모든 Linestring이 아닌 직접적으로 맞닿아 있는 일부 Linestring(l_1 과 l_2)에 대해서만 G1 연속성을 확보하는 방식으로 최적화를 진행하면 된다.

즉, 지도 전체를 다시 최적화할 필요 없이, 인접한 클러스터들이 맞닿아 있는 일부 Linestring들만을 대상으로 소규모 최적화 문제를 해결함으로써 클러스터 간 매개변수화 결과를 효율적으로 융합할 수 있다.

이를 위해, 인접한 두 클러스터가 만나는 구간에서 모든 연결이 단순한 직렬 연결이 되도록 도로 네트워크 그래프를 분할하는 방법을 제안한다. 여기에서 핵심은 Fig. 4에서 l_1 과 l_2 처럼 단순 직렬로 연결된 Linestring들은 서로 다른 클러스터에 배치하고, 주황색 점을 공유하는 여러 개의 Linestring들은 동일한 클러스터에 포함시키는 것이다.

이러한 원칙을 반영하여, 클러스터 생성 규칙을 다음과 같이 설정하였다.

1. 특정 노드를 세 개 이상의 Linestring이 공유하는 경우, 해당 Linestring들은 모두 같은 클러스터에 속한다.
2. 특정 노드를 두 개의 Linestring만이 공유하더라도, 이들이 하나의 경로 상에 있지 않다면 같은 클러스터에 포함된다.
3. 하나의 Linestring은 서로 다른 클러스터에 속할 수 없으며, 두 클러스터가 동일할

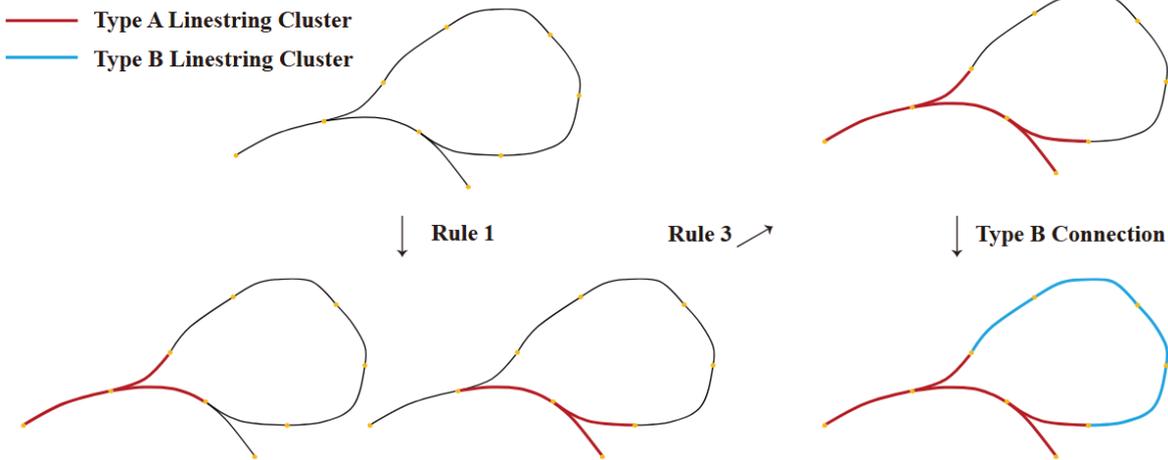


Fig. 5 도로 네트워크 분할 예시. 규칙에 따라 생성된 클러스터는 Type A 클러스터이고, 그 외 나머지 Linestring들은 연결관계에 따라 모여서 Type B 클러스터가 된다.

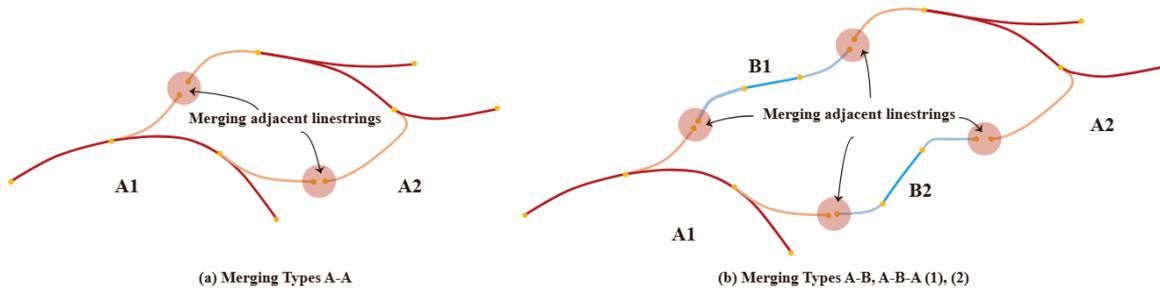


Fig. 6 Linstreing 클러스터 종류(A,B)에 따른 융합 방법. Type A와 A 클러스터가 인접해 있는 경우(a)와 Type A와 B가 섞여서 연결되어 있는 경우(b) 모두 직접적으로 맞닿아 영역(빨간색 원)에 대해서만 소규모 최적화 문제를 풀어서 G1 연속성을 확보하면 된다.

Linstreing을 공유할 경우 하나의 클러스터로 병합된다.

된 지도 내용을 갱신할 때도 효율적으로 적용할 수 있음을 보인다.

위의 규칙들을 적용하면 주어진 차선들을 작은 단위의 독립적인 클러스터로 나눌 수 있으며, 그 예시는 Fig. 5에 나타나 있다. 전체 지도가 주어졌을 때, 먼저 모든 노드에 대해 규칙 1을 적용하고, 해당 규칙의 영향을 받은 차선들을 그림에서 빨간색으로 표시하였다.

이 과정에서 두 개의 노드가 규칙 1의 영향을 받았으며, 이에 따라 각 노드에 연결된 Linstreing들은 하나의 클러스터에 속해야 한다. 그러나 이 두 그룹이 하나의 Linstreing을 공유하고 있기 때문에, 규칙 3에 따라 다시 하나의 클러스터로 병합된다.

이러한 과정을 통해 생성된 클러스터를 Type A 클러스터로 명명하며, Type A 클러스터에 속하지 않는 나머지 차선(Linstreing)들은 Fig. 5에서처럼 서로 연결된 성분들끼리 묶여 Type B 클러스터를 형성한다.

이와 같이 구성된 클러스터들은 Fig. 4에서 설명한 것처럼, 인접한 클러스터 간의 연결 부분에서 항상 Linstreing들이 단순 직렬 형태로 정렬된다. 따라서, 전역적인 최적화 없이도 소규모 최적화만으로 인접한 클러스터들을 G1 연속하게 연결할 수 있으며, 이를 통해 연산 과정의 효율성을 획기적으로 향상시킬 수 있다.

3.2 인접한 Linstreing 클러스터 융합 방법

이번 절에서는 앞서 생성한 Linstreing 클러스터들의 원호 매개변수화 결과(2.3절 적용)를 G1 연속하게 융합하는 과정에서, 전체 Linstreing을 고려할 필요 없이 직접적으로 맞닿아 있는 일부 Linstreing에 대해서만 소규모 최적화를 수행하면 충분함을 설명한다. 또한, 이를 통해 매개변수화

3.2.1 클러스터 종류에 따른 융합 방법

앞서 3.1.2절에서 제안한 그래프 분할 기법을 활용해 Linstreing 클러스터를 생성하고, 각 클러스터를 원호로 매개변수화(2.3절 적용)한 후에는 인접한 클러스터 간 G1 연속성이 유지되도록 융합해야 한다. 이때, 클러스터의 유형에 따라 융합 방식이 조금씩 달라지며, 크게 Type A와 A 간의 융합과 Type A와 B 간의 융합으로 구분된다. 특히, Type A와 B 간의 융합은 세 가지 유형으로 세분화된다.

Type A와 A 간의 융합은 Fig. 6(a)에서 볼 수 있듯이, 여러 지점에서 연결되더라도 빨간색 원으로 표시된 영역 내에서만 소규모 최적화를 수행하면 되므로, 각 빨간색 원마다 독립적으로 최적화를 진행한다. 여기서 말하는 소규모 최적화는 Fig. 4에서 설명한 것처럼, 직접적으로 맞닿아 있는 일부 Linstreing들에 대해서만 제한적으로 수행하는 것이다.

Type A와 B 간의 융합은 Fig. 6(b)에서처럼 세 가지 유형으로 나뉜다. 클러스터 A1과 A2가 다를 경우 해당 융합을 A-B-A (1), A1과 A2가 사실상 하나의 클러스터일 경우 A-B-A (2)로 지정한다. 만약 A1 또는 A2 중 하나가 존재하지 않는 경우에는 A-B 유형으로 명명한다. 하지만 이러한 유형의 차이에도 불구하고, 융합 과정은 본질적으로 Type A와 A 간의 융합과 동일하다. 즉, 모든 경우에서 빨간색 원으로 표시된 영역 내에서 소규모 최적화를 수행하여 인접한 클러스터 간 G1 연속성을 효과적으로 확보한다.

참고로, 빨간색 원으로 표시된 소규모 최적화는 2.3절에서 소개한 다중 Linstreing 매개변수화 알고리즘을 그대로 적용하되, 전체 지도에 대

해 수행하는 것이 아니라 직접 맞닿아 있는 소수의 Linestring들에만 국한하여 적용한다. 이를 통해 전역적인 최적화 없이도 인접한 클러스터 간 G1 연속성을 유지하면서 연산 효율성을 크게 향상시킬 수 있다.

3.2.2 기존 그래프 분할 방법에 대한 고찰

앞서 3.1.1절에서 기존 Louvain 방법 및 전통적인 그래프 분할 기법이 전체 지도에 대한 최적화를 필연적으로 수반하여 연산 효율성이 떨어진다고 언급하였는데, 이를 본 연구에서 제안한 알고리즘과 비교하여 보다 구체적으로 설명하고자 한다.

본 연구에서 제안한 그래프 분할 기법은 Linestring 클러스터들이 만나는 지점이 항상 단순 직렬 연결되도록 강제하는 방식이다. 반면, 기존 그래프 분할 기법들은 클러스터 간 연결의 구조를 규격화하는 것이 아니라, 단순히 클러스터 간의 연결 자체를 최소화하는 데 초점을 맞추고 있어 결과적으로 생성되는 클러스터의 형태가 크게 달라진다.

이러한 차이로 인해, 기존 그래프 분할 방법을 적용하면 클러스터 간의 연결이 복잡하게 형성될 가능성이 높으며, 따라서 클러스터를 융합할 때 소규모 최적화를 적용하기 어렵다. 결국, 2.4절에서 언급한 바와 같이 지도 전체에 대한 최적화를 피할 수 없고, 이로 인해 연산 병목이 해소되지 않는 문제가 발생한다.

3.2.3 매개변수화된 지도의 갱신 방법

지금까지 소개한 연구는 노이즈가 포함된 차선 데이터와 차선 간의 연결 정보를 활용하여 이를 원호로 매개변수화하는 방법을 다루었다. 자율주행 및 ADAS에서 정밀지도 데이터를 실시간으로 활용하려면 지속적인 지도 갱신이 필수적인데, 본 연구에서 제안한 기법을 적용하면 전체 지도를 다시 최적화할 필요 없이, 변경이 발생한 Linestring 클러스터들만 선택적으로 최적화할 수 있다. 이를 통해 지도 갱신의 효율성을 크게 향상시킬 수 있다.

4. 알고리즘 검증 및 분석

본 장에서는 2장과 3장에서 제안한 그래프 분할 기반 차선 지도 원호 매개변수화 알고리즘을 실제 데이터셋에 적용하여 성능을 검증하고 분석한다.

4.1 데이터셋 및 구성

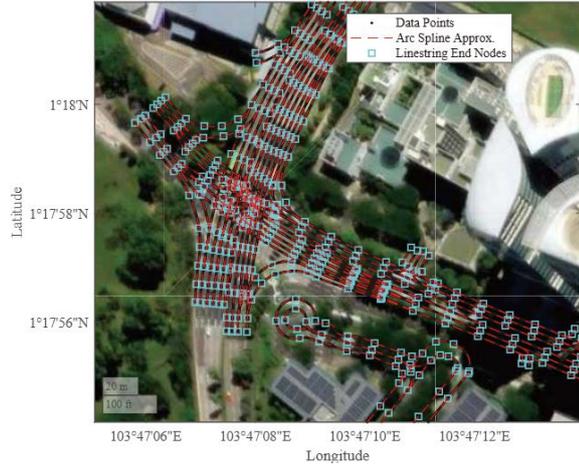


Fig. 7 차선 지도 원호 매개변수화 결과(교차로 구간)

제안된 알고리즘을 검증하기 위해, 자율주행 연구에서 널리 활용되는 nuScenes 데이터셋의 정밀지도를 Lanelet2⁵⁾ 형식으로 변환한 연구 내용¹⁷⁾을 사용하였다. 보스턴과 싱가포르의 일부 지역을 포함한 총 4개의 지도에 대해 원호 매개변수화를 수행하는데, 각각 Seaport(Boston), Queenstown(Singapore), One North(Singapore), Holland Village(Singapore) 지역을 포함한다. 이후 결과를 정리한 표에서 BS는 Boston Seaport, QT는 Queenstown, ON은 One North, HV는 Holland Village를 의미한다.

실험은 Intel i7-12700 CPU, 32GB RAM을 갖춘 PC에서 진행되었으며, Lanelet2 형식의 지도로부터 차선 데이터를 추출하는 과정은 Python에서, 원호 매개변수화 및 도로 네트워크 그래프 분할을 포함한 최적화 과정은 MATLAB에서 실행되었다.

참고로, 대규모 지도에 대한 매개변수화는 이전에 진행된 연구가 없기 때문에, 본 연구에서 새롭게 제안된 내용에 대한 분석과, 기존 지도의 형식인 Polyline과 비교하였다.

4.2 결과

본 절에서는 제안된 알고리즘에 대한 정성적 및 정량적 분석을 수행한다. 구체적으로, 제안된 도로 네트워크 분할 기법을 통해 생성된 Linestring 클러스터의 특성을 분석하고, 원호 매개변수화 결과를 평가하며, 기존 그래프 분할 기법과의 성능을 비교하였다.

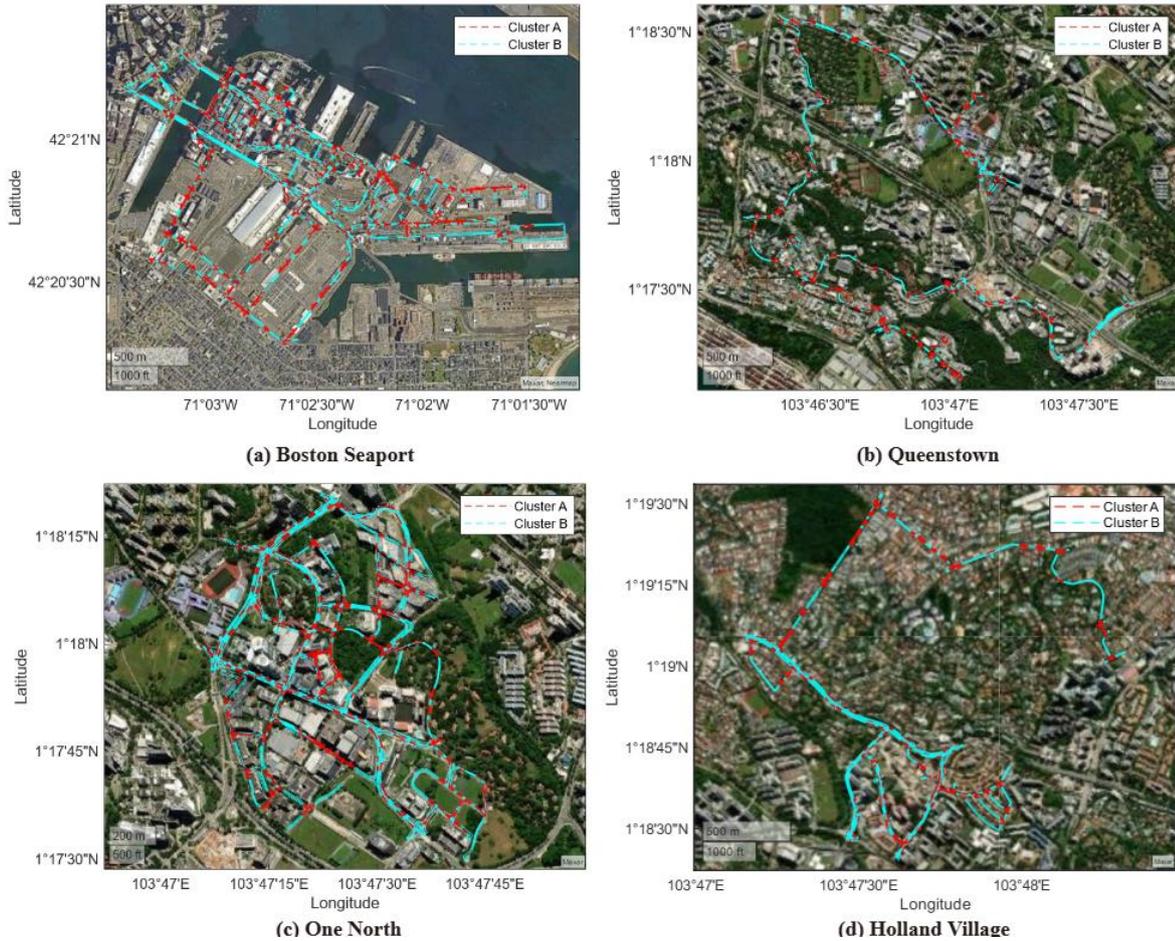


Fig. 8 총 4개의 지도에 대한 클러스터 형성 결과. 대체로 Type A 클러스터는 복잡하게 연결되어 있는 교차로 구간을, Type B 클러스터는 단순하게 연결되어 있는 구간을 나타내는 것을 확인할 수 있다.

Metrics	P _{0.03}	P _{0.05}	P _{0.07}	AP	RMSE(m)
BS	45.123	79.697	94.390	73.064	0.0426
QT	45.744	77.452	92.952	72.049	0.0416
ON	45.593	78.067	93.545	72.401	0.0419
HV	45.408	78.557	93.718	72.561	0.0410

Table 2. 4가지 지도에 대한 원호 매개변수화 근사 RMSE 및 정밀도(Precision)

4.2.1 Linestring 클러스터 생성 결과

Table 1에서 확인할 수 있듯이, 총 4개의 지도에서 도로 네트워크 그래프를 분할한 결과, Linestring이 Type A 또는 Type B 클러스터로 자동으로 분류되었다. 또한, 각 클러스터의 유형에 따라 융합 방식도 달라지는 것을 확인할 수 있다.

전체 지도에서 Type A와 B 클러스터의 공간적 분포는 Fig. 8에서 시각적으로 확인할 수 있다. 그림을 살펴보면, Type A 클러스터는 주로 복잡

Maps	BS	QT	ON	HV
Number of Linestrings	9273	6253	6311	4078
Number of Points	662523	430789	460630	297042
Storage Size (Points)	1325046	861578	921260	594084
Number of Arc Nodes	16360	11271	10658	7312
Number of Arc Segments	17439	12037	11195	7690
Storage Size (Arcs)	67598	46616	43706	30004
Storage Ratio(Points/Arcs)	19.602	18.483	21.079	19.800

Table 3. 4가지 지도에 대한 원호 매개변수화 저장 효율 비교표(기존 Polyline 형식과의 비교)

하게 연결된 교차로 영역에 집중되어 있으며, Type B 클러스터는 교차로 간 단순한 연결 구간을 주로 포함하고 있음을 알 수 있다.

4.2.2 원호 매개변수화 결과

원호를 이용한 점군 데이터의 근사 오차 분포는 Table 2에서, 저장 효율 향상에 대한 분석은 Table 3에서 확인할 수 있다.

Table 2에 따르면, 원호 근사의 RMSE는 평균적으로 4.2cm로 나타나 상당히 높은 정확도를 보인다. 또한, 오차 분포의 극단적인 사례 여부를 분석하기 위해 정밀도(Precision)를 활용한 추가 분석을 수행하였다. 원호와 특정 점 사이의 거리를 오차로 정의하고, 이 값이 설정된 임계값 t 이하일 경우 참(True), 초과할 경우 거짓(False)으로 분류한 후, 전체 점 개수 대비 참의 비율을 정밀도로 계산하였다. 이를 임계값 $t = 0.03, 0.05, 0.07$ 등으로 변화시키며 측정 후, 평균을 구한 값은 Average Precision(AP)으로 정의된다.

Table 3을 보면, 평균적으로 $t = 0.07$ 기준 정밀도가 93%로 나타났으며, 이는 전체 점군 데이터 중 93%가 7cm 이하의 원호 근사 오차를 갖는다는 것을 의미한다. 또한, 네 개의 지도에 대한 정밀도를 비교한 결과, 오차 분포가 극단적이지 않고 평균을 중심으로 균일하게 분포하는 경향을 보여, 제안된 원호 매개변수화 알고리즘이 안정적인 성능을 제공할 수 있음을 확인할 수 있다.

Table 3의 결과를 기반으로 기존 Polyline 형식과 비교했을 때, 원호를 이용하여 차선 정보를 저장하면 약 20배의 저장 공간 절약 효과가 있음을 확인할 수 있다. 물론, 이는 기존 지도가 점 데이터를 저장하는 방식에 따라 차이가 발생할 수 있지만, 원호 기반의 방식은 단순한 데이터 압축 이상의 의미를 가진다. 즉, 점군 데이터를 보다 효율적으로 저장하면서도, 원호라는 기하학적 표현을 통해 자율주행 및 ADAS에서 유용한 도로 곡률 정보 등을 제공할 수 있다는 점에서 높은 실용성을 갖는다.

4.2.3 기존 그래프 분할 기법과의 비교

마지막으로, 2.3절에서 제안한 원호 매개변수화 알고리즘과 기존의 그래프 분할 기법(Graph Component, Louvain 방법)을 결합하여, 3장에서 제안한 도로 네트워크 분할 알고리즘과 성능을 비교하였다. 앞서 3.2.2절에서 설명했듯이, 기존의 Graph Component나 Louvain 방법은 전체 지도 단위의 최적화가 필수적이므로, 4개의 지도에 대해 매개변수화를 실행했을 때 하루 이상의 시간이 지났음에도 불구하고 몇 차례의 iteration만 진행되는 등 최적화 속도가 매우 느렸다. 이로 인해 실행 시간을 측정하는 것이 현실적으로 어려워 강제 중단하였다.

반면, 본 연구에서 제안한 도로 네트워크 분할 기법을 2.3절의 원호 매개변수화 알고리즘과 결합한 경우, 평균적으로 1시간에서 1시간 30분 이내에 최적화가 완료되었다. 물론, 실행 시간

은 사용된 PC의 성능에 따라 달라질 수 있지만, 본 연구는 전역적인 G1 연속성을 유지하면서도 현실적인 시간 내에 차선 매개변수화를 수행할 수 있음을 최초로 입증했다는 점에서 중요한 의미를 갖는다.

5. 결론

본 연구에서는 복잡하게 연결된 대규모 정밀지도의 차선 데이터를 원호로 매개변수화하는 방법을 제안하고, G1 연속성을 유지하면서도 효율적으로 매개변수화를 수행할 수 있도록 도로 네트워크 분할 기법을 새롭게 개발하였다. 제안된 알고리즘은 대규모이면서 복잡한 도로 네트워크를 매개변수화한 최초의 연구라는 점에서 중요한 의미를 가지며, 실제 자율주행 데이터셋의 정밀지도를 활용하여 그 성능을 검증하였다.

본 연구의 결과는 자율주행 시스템과 연동될 경우, 정밀지도의 지속적인 갱신과 확장을 보다 효율적으로 수행할 수 있도록 기여할 것으로 기대된다.

후기

이 연구는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원 (No. RS-2024-00346702), 산업기술혁신사업 캐빈교체형 자율서플 공용새시플랫폼 기술 개발 연구비 지원 (20014983), 정부(산업통상자원부 및 한국산업기술평가관리원)의 재원으로 자율주행기술개발혁신사업단의 지원(20018181)을 받아 수행된 연구임.

References

- 1) Endo, Y., Javanmardi, E., Gu, Y. et al. High Definition Map Aided Object Detection for Autonomous Driving in Urban Areas. *Int.J Automot. Technol.* 24, 469–481 (2023). <https://doi.org/10.1007/s12239-023-0039-0>
- 2) “TomTom HD Map with RoadDNA,” <https://www.tomtom.com/products/hd-map/>, 2018, accessed: 2024-09-09.
- 3) “Here HD Live Map - On the Road Towards Autonomous Driving,” <https://www.here.com/platform/HD-live-map>, 2018, accessed: 2024-09-09.
- 4) M. Haklay and P. Weber, “OpenStreetMap: User-Generated Street Maps,” *IEEE Pervasive Computing*, vol. 7, no. 4, pp. 12–18, 2008.
- 5) F. Poggenhans, J.-H. Pauls, J. Janosovits, S. Orf, M. Naumann, F. Kuhnt, and M. Mayr, “Lanelet2: A high-definition map framework for the future of automated driving,” in 2018 21st International Conference on Intelligent Transportation Systems (ITSC), 2018, pp. 1672–1679.

- 6) G.-P. Gwon, W.-S. Hur, S.-W. Kim, and S.-W. Seo, "Generation of a Precise and Efficient Lane-Level Road Map for Intelligent Vehicle Systems," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 6, pp. 4517–4533, 2017. pp.3247–3256.
- 7) C. Guo, K. Kidono, J. Meguro, Y. Kojima, M. Ogawa, and T. Naito, "A Low-Cost Solution for Automatic Lane-Level Map Generation Using Conventional In-Car Sensors," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 8, pp. 2355–2366, 2016.
- 8) L. Qiao, W. Ding, X. Qiu, and C. Zhang, "End-to-End Vectorized HD-Map Construction With Piecewise Bezier Curve," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 13 218–13 228.
- 9) T. Zhang, S. Arrigoni, M. Garozzo, D. ge Yang, and F. Cheli, "A lane-level road network model with global continuity," *Transportation Research Part C: Emerging Technologies*, vol. 71, pp. 32–50, 2016.
- 10) K. Jo and M. Sunwoo, "Generation of a Precise Roadway Map for Autonomous Cars," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 3, pp. 925–937, 2014.
- 11) B. Gallazzi, P. Cudrano, M. Frosi, S. Mentasti, and M. Matteucci, "Clothoidal Mapping of Road Line Markings for Autonomous Driving High-Definition Maps," in *2022 IEEE Intelligent Vehicles Symposium (IV)*, 2022, pp. 1631–1638.
- 12) S. Zhang, R. Wang, Z. Jian, W. Zhan, N. Zheng, and M. Tomizuka, "Clothoid-based reference path reconstruction for hd map generation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 1, pp. 587–601, 2024.
- 13) A. Schindler, G. Maier, and F. Janda, "Generation of high precision digital maps using circular arc splines," in *2012 IEEE Intelligent Vehicles Symposium*, 2012, pp. 246–251.
- 14) J. Jeon, Y. Hwang, and S. B. Choi, "Reliability-based G1 continuous arc spline approximation," *Computer Aided Geometric Design*, vol. 112, p. 102363, 2024.
- 15) V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, p.P10008, oct 2008.
- 16) M. E. J. Newman, "Modularity and community structure in networks," *Proceedings of the National Academy of Sciences*, vol. 103, no. 23, pp. 8577–8582, 2006.
- 17) A. Naumann, F. Hertlein, D. Grimm, M. Zipfl, S. Thoma, A. Rettinger, L. Halilaj, J. Luetin, S. Schmid, and H. Caesar, "Lanelet2 for nuScenes: Enabling Spatial Semantic Relationships and Diverse Map-Based Anchor Paths," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2023,