

안전 인프라 데이터를 활용한 안전 경로 추천 및 시각화 시스템

김찬진⁰, 신동화
광운대학교 컴퓨터정보공학부
khlk0713@naver.com, dhshin@kw.ac.kr

Safety Path Recommendation and Visualization System Using Safety Infrastructure Data

ChanJin Kim⁰, DongHwa Shin
School of Computer and Information Engineering, Kwangwoon University

요 약

본 연구에서는 도시 내 안전 인프라 데이터를 활용하여 안전 기반 경로를 최적화하고 시각화하는 시스템을 제안한다. 최근 도시 환경에서 범죄 예방 및 시민 안전에 대한 관심이 증가함에 따라, 기존의 최단 경로 중심의 경로 탐색 알고리즘에 안전 요소를 반영할 필요성이 대두되고 있다. 본 시스템은 도시 곳곳에 설치된 CCTV 및 가로등 데이터를 기반으로 각 경로의 안전도를 정량화하여 최단 경로와 안전 경로를 동시에 제공하는 것을 목표로 한다. 이 시스템은 서버와 실시간 통신을 수행하고, 지도 상에 경로 및 주변 안전 인프라(예: CCTV, 가로등) 정보를 활용하여 시각화한다. 경로 탐색 시 두 경로를 다양한 차트를 통해 비교 및 분석할 수 있도록 한다. 본 연구에서 제안한 시스템은 실시간으로 도시 안전 데이터를 반영하여 최적 경로를 제공할 뿐 아니라, 다양한 시각화 도구를 통해 사용자에게 직관적이고 이해하기 쉬운 위험도 정보를 전달할 수 있다. 본 연구는 도시 내 경로 탐색 시 안전성을 고려한 의사결정 지원 시스템의 가능성을 제시하며, 향후 다양한 안전 지표와 교통 데이터를 통합한 종합적인 도시 안전 최적화 시스템으로 확장될 수 있는 기반을 마련한다.

1. 서 론

급격한 도시화와 인구 증가에 따라 도시 내 교통 혼잡, 범죄 발생률, 사고 등 다양한 안전 문제가 대두되고 있다. 전통적인 경로 탐색 알고리즘은 주로 최단 시간이나 최단 거리를 기준으로 경로를 산출하지만, 최근에는 시민의 안전을 보장하는 것이 중요한 사회적 요구로 대두되면서, 경로 선택 시 안전 요소를 함께 고려하는 새로운 접근법이 필요해졌다. 특히 도시의 안전 인프라인 CCTV, 가로등 등은 범죄 예방 및 응급 상황 대응에 기여하며, 이들 데이터는 경로의 안전도를 정량화할 수 있는 중요한 지표로 활용될 수 있다.

본 연구에서는 공공 데이터[1]를 기반으로, 기존의 최단 경로와 더불어 안전성을 고려한 경로(이하 '안전 경로'라 한다.)를 동시에 제공하고, 이를 지도 및 다양한 시각화 도구를 통해 사용자에게 직관적으로 전달하는 시스템을 제안한다. 기존 경로 탐색 시스템이 주로 이동 거리나 소요 시간에 초점을 맞춘데 반해, 본 연구는 사용자의 안전까지 고려한 경로 선택을 지원하는 것을 목표로 한다. 이를 위해 도시 내 안전 인프라 데이터를 수집하고, 경로별 안전도를 정량화하여 Leaflet[2] 기반 지도 시각화 및 HTML Canvas 기반 그래픽 요소로 표현하였다. 제안하는

시스템은 최단 경로와 안전 경로를 지도상에 동시에 표시하고, 게이지 차트 및 막대 그래프를 활용하여 경로 간 위험 비율과 안전도 차이를 명확히 시각화함으로써 사용자가 각 경로의 특성을 한눈에 비교할 수 있도록 지원한다. 본 논문에서는 제안한 시스템의 설계 및 구현 과정을 상세히 기술하고, 실제 도시 데이터를 활용한 실험을 통해 시스템의 유효성과 실용성을 평가함으로써, 경로 탐색 시 안전성 고려를 위한 의사 결정 지원 시스템으로서의 가능성을 모색하고자 한다.

2. 최단 경로 및 안전 경로 알고리즘 구현

본 연구에서는 도로망 그래프 상에서 최단 경로(shortest path)와 안전 경로(safe path)를 동시에 고려하기 위한 알고리즘을 구현하였다. Python의 NetworkX 라이브러리[3]를 활용하여 전체 도시 구역을 노드와 엣지로 표현하고, CCTV·가로등 정보를 별도로 추출하여 그래프에 특성 노드로 추가하였다. 각 노드의 속성을 "cctv", "streetlight", "seoul"로 나누어서 부여해 경로 탐색 시 서로 다른 가중치 계산에 사용되어, 단순히 최단 거리를 추구하는 경우와 안전도를 반영하는 경우를 구분한다.

먼저, 최단 경로 알고리즘은 엣지의 물리적 거리를 가중

치로 하여, 시작점에서 도착점까지 이동 거리가 최소가 되는 경로를 탐색한다. 이를 위해 Haversine 공식[4]을 이용해 두 노드 간의 위·경도 좌표로부터 실제 거리를 계산하고, NetworkX를 통해 최단 경로를 얻는다. 이때, CCTV나 가로등에 대한 고려 없이 단순 거리만을 기준으로 경로를 선정하기 때문에 이동 소요 시간이 가장 짧은 경로를 찾는 효과를 기대할 수 있다.

반면, 안전 경로 알고리즘은 CCTV와 밀집도가 높은 구역을 선호하도록 가중치를 재조정한다. CCTV와 가로등 노드가 인접한 구역("seoul" 속성 노드)일수록 경로 비용을 의도적으로 낮추는 방식을 택한다. 이를 공식화하기 위해, 아래와 같은 가중치 함수(식 1)를 사용한다. 노드 u 가 일반 도로("seoul")일 경우, 두 거리의 값 중 더 큰 값($\max(d(u,v)*r, d(u,v)-\alpha...)$)을 선택하는 데 각각의 가중치 계산은 이와 같다. 주변 CCTV·가로등 개수($CCTV(u)+SL(u)$)에 따라 가중치(α)를 감소시키고, 너무 작아지는 상황을 방지하기 위해 일정 비율(λ) 이하로는 내려가지 않도록 설계하였다.

$$W_{safe}(u,v) = \begin{cases} \max(d(u,v) \times \gamma, d(u,v) - \alpha \cdot [CCTV(u) + SL(u)]) \\ d(u,v) \text{ if } shortestpath \end{cases}$$

식 1 Safe Weight Function

노드 u 가 일반 도로 노드인 경우에만 주변 CCTV 및 가로등 수에 따라 가중치를 낮춰주어 그 구간을 "보다 짧은 거리"로 인식하게 된다. 반면 CCTV나 가로등 노드에 대해서는 안전 가중치 보정을 적용하지 않고 원본 거리를 그대로 사용한다. 결과적으로 CCTV와 가로등이 풍부한 길은 가중치가 작아져 안전 경로로 우선 선택이 될 가능성이 높아지는 구조이다.

3. 시각화 기법

본 연구에서는 도시 안전 데이터를 기반으로 경로의 안정성을 정량화하고, 이를 사용자에게 직관적으로 전달하기 위해 웹 기반 애플리케이션으로서 잘 알려진 웹 프레임워크인 React[5]를 기반으로 구현되었다. 주요 시각화 도구로는 막대그래프, 히스토그램, 게이지 차트가 있으며, 각각의 기법은 다음과 같이 구현되었다.

3.1 인프라 분포 히스토그램

히스토그램(그림 1)은 경로 데이터(예: 각 데이터 포인트 마다의 CCTV와 가로등 수)를 바탕으로, x축은 경로상의 데이터 포인트 수, y축은 안전 인프라의 분포 및 값을 나타내도록 구현되었다(그림 1). 전체로, x축은 경로상의 데이터 포인트 수, y축은 안전 인프라의 분포 및 값을 나타내도록 구현되었다(그림 1). 전체 데이터 중 최댓값을 기반으로 각 막대의 높이를 결정하며, 기준선을 중심으로

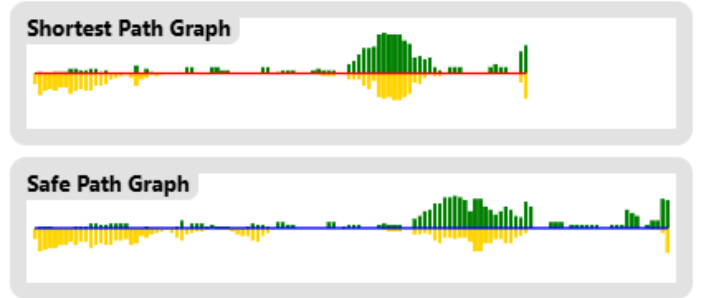


그림 1 경로 구간별 인프라 데이터를 표시한 히스토그램

상단은 CCTV(녹색), 하단은 가로등(노란색) 히스토그램으로 분리하여 표현하였다.

3.2 막대 그래프

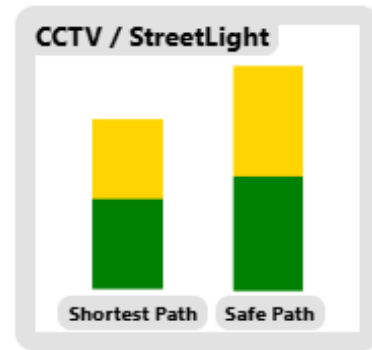


그림 2 경로의 인프라 데이터 수를 나타낸 막대 그래프

막대 그래프(그림 2)는 각 경로 별 CCTV(녹색) 및 가로등(노란색)의 수를 합하여 나타낸다. 전체 데이터 중 최댓값을 기준으로 각 막대의 높이를 산출하므로, 서로 다른 경로의 안전 인프라 수치가 사이하더라도 비교가 가능하다.

3.3 위험 게이지

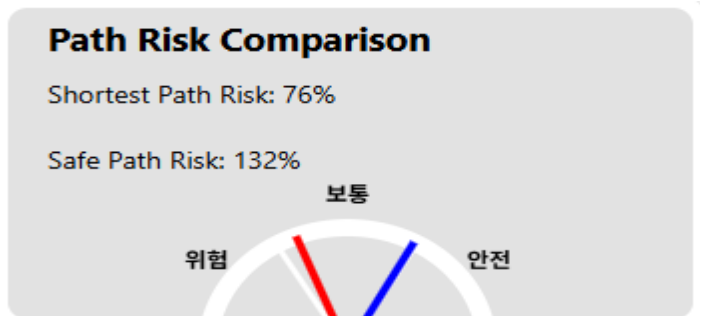


그림 3 경로 별 속한 위험 구간 및 위험 게이지

위험 게이지(그림 3)는 사용자가 경로의 위험도를 한눈에 파악할 수 있도록 Canvas를 활용하여 반원 형태의 게이지를 그린다. 안전 경로와 최단 경로의 위험도를 비교해 백분율로 환산한다. 이 반원 게이지는 위험, 보통, 안전의 세 구간으로 나뉘며, 위험도에 따라 바늘이 그려진다. 최단 경로는 빨간색, 안전 경로는 파란색이며 두 경로 간의

안정성 차이를 직관적으로 파악할 수 있도록 지원한다.

3.4 지도 상의 안전 인프라 마커 표시

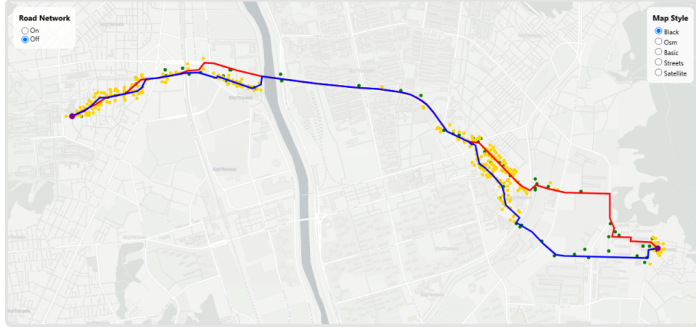


그림 4 두 경로가 동시에 표시된 지도

경로 탐색 결과(최단 경로: 빨간색, 안전 경로: 파란색)와 함께 지도 상(그림 4)에 안전 인프라 마커를 표시한다. 서버로부터 수신한 데이터를 기반으로, 각 노드별 CCTV와 가로등의 ID 배열을 추출한다. 이때, 경로에 포함된 노드에 해당하는 ID들만을 별도의 배열로 구성한다. Leaflet 기반 지도 위에 필터링된 CCTV와 가로등 마커를 표시한다. 이 Layer는 지도에 오버레이(overlay)되어, 각 마커의 위경도 좌표를 canvas 내 좌표로 변환 후 원(CCTV: 녹색, 가로등: 노란색)으로 그려진다. 사용자로부터 새로운 경로 요청이 발생할 때마다 새로운 경로 탐색 결과 및 마커 데이터로 업데이트되어 사용자에게 명확한 정보를 제공한다.

3.5 사용자 맞춤 및 반응형 인터페이스

그림 5 경로 탐색을 위한 사용자 인터페이스

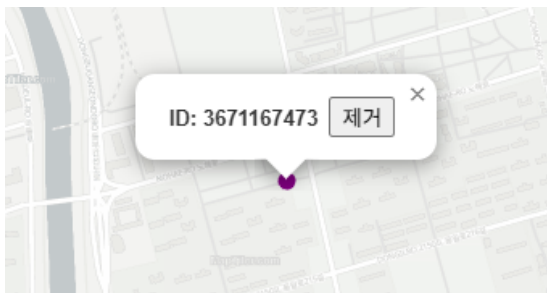


그림 6 노드 마커 클릭 시 표시되는 정보

본 인터페이스(그림 5)는 사용자가 지도에 한 부분을 클

릭하여 마커를 생성하고 그 마커에 대한 정보(그림 6)를 Start, End Node로 입력한 후 슬라이더를 이용해 안전 가중치(safe weight)를 조절할 수 있도록 하고, 이 값에 따라 경로 탐색 결과와 시각화 요소(히스토그램, 막대 그래프, 위험 게이지, 인프라 마커)가 실시간으로 업데이트 되도록 하였으며, 각 색깔이 무엇을 나타내는지에 대한 설명으로 사용자가 시각화 요소들을 이해하기 쉽게 설계되었다.

4. 결론 및 향후 연구

본 연구에서는 도시 안전 인프라 데이터를 기반으로 경로의 안정성을 정량화하고, 이를 다양한 시각화 도구를 통해 사용자에게 직관적으로 전달하는 시스템을 제안하였다. 구체적으로, CCTV와 가로등 데이터를 활용하여 최단 경로와 안전 경로의 위험도를 산출하였으며, 사용자가 슬라이더를 통해 안전 가중치를 조절할 수 있도록 하여 경로 탐색 시 안정성과 최단 경로 간의 트레이드-오프를 고려할 수 있는 실시간 의사결정 지원 체계를 구현하였다.

그러나 본 연구에는 다음과 같은 한계점이 존재한다. 경로 선택에 안전 요소만을 반영했으므로, 실제 이용자 입장에서는 교통량, 소요시간, 도로 혼잡도 등 다른 요소와의 종합적인 고려가 필요하다. 또한, 경로 산출에 사용되는 가중치 함수가 극히 일부 경로에서 최단 경로가 안전 경로보다 더 안전도가 높고 경로도 길게 나오는 경우가 있었다. 향후 연구에서는 위의 한계들을 개선하기 위해 다음과 같은 방안을 고려할 수 있다. 첫째, 다양한 안전 지표를 통합하는 것이다. 현재는 CCTV와 가로등 데이터를 중심으로 위험도를 산출하였으나, 향후 교통량, 사고 발생 이력, 범죄 통계 등 다양한 안전 지표를 통합하여 종합적인 위험 분석 모델을 개발할 수 있다. 둘째, 데이터에 대한 신뢰성을 보완하는 것이다. 현재 가중치를 부여하는 부분에 대해 보완하는 알고리즘을 도입하여, 각 경로에 대한 정확도를 높일 수 있다. 셋째, 시각화 기법을 보완하고 확장하는 것이다. Canvas 기반의 시각화 외에도, D3.js, Chart.js 등 고급 시각화 라이브러리를 적용하여 반응형 및 상호작용이 강화된 시각화 도구를 개발할 수 있다.

이와 같이 본 연구는 도시 안전 데이터를 활용한 위험 기반 경로 최적화 및 시각화 시스템의 가능성을 확인하였으며, 향후 다양한 안전 지표와 고도화된 시각화 기법을 통합함으로써, 실제 도시 안전성을 고려한 스마트 경로 안내 시스템으로 발전시킬 수 있을 것으로 기대한다.

참고 문헌

- [1] 서울열린데이터광장, <https://data.seoul.go.kr/>
- [2] Leaflet: An open-source JavaScript library for interactive maps, Leaflet, <https://leafletjs.com>
- [3] NetworkX: Network Analysis in Python, NetworkX, <https://networkx.org/>
- [4] R. W. Sinnott, "Virtues of the Haversine," *Sky and Telescope*, vol. 68, no. 2, p. 159, 1984.
- [5] React 공식 문서, Meta, <https://ko.react.dev/>