

바이러스의 빠른 검사를 위한 도시 맵과 K-d 트리 기반의 그래프 구축과 효율적인 환자 이동 경로 시스템

신영찬⁰, 문성혁*, 김동희*, 김종현*
⁰강남대학교 소프트웨어응용학부,
 *강남대학교 소프트웨어응용학부
 e-mail: jonghyunkim@kangnam.ac.kr

Quick Inspection of Virus Using Urban-Map and K-d Tree based Graph Construction and Efficient Patient Movement Route System

Young Chan Shin⁰, Seong-Hyeok Moon*, Donghui Kim*, Jong-Hyun Kim*
⁰School of Software Application, Kangnam University,
 *School of Software Application, Kangnam University

● 요약 ●

본 논문에서는 COVID-19와 같은 위급한 상황에서 바이러스 검사를 빠르게 진행하기 위한 K-d 트리 기반의 그래프 구축과 환자 이동 경로 시스템을 제안한다. 가상환경에서 활용되는 대표적인 길 찾기 알고리즘은 A*나 NavMesh 자료구조는 정해진 정적 이동 경로만을 안내하려는 방법이기 때문에 가상환경에서 NPC를 제어할 때는 효율적이지만, 실제 환경에 적용하여 문제를 풀기에는 충분하지 않다. 특히, 빠른 바이러스 검사를 받기 위해서는 짧은 거리만을 이용하는 게 아닌, 실제 도로 교통상황, 병원의 크기, 환자 이동 수, 환자 처리 시간 등 고려해야 할 사항들이 많다. 본 논문에서는 위에서 언급한 다양한 속성들과 이를 이용한 최적화 함수를 모델링하여, 실제 도시 맵에서 바이러스 검사를 빠르고 효율적으로 제어할 수 있는 프레임워크를 제안한다.

키워드: 바이러스 검사(Virus inspection), 길 찾기(Pathfinding), 그래프 구축(Graph construction), 이동 경로(Motion trajectory), 도시 맵(Urban-map)

I. Introduction

신종플루, 메르스, COVID-19처럼 바이러스는 인간과 동물을 지속적으로 위협하고 있으며, 현대화 사회에서 기술적인 진보가 있었음에도 불구하고 바이러스는 여전히 위협적인 존재이다. COVID-19의 확산 이후 2년이 지난 현재까지도 전염성이 매우 빨라 쉽게 주변 사람들에게 퍼지며 국가 위기 경보까지 될 수 있는 상황이 나타나고 있다. 지역감염이나 N차 감염을 예방하기 위해서는 사람들이 이른 시일 안에 검사를 받고 치료 및 예방하는 행동이 필요하지만, 현재는 단순하게 문자로만 정보를 제시하며 정확한 가이드 시스템이 없다.

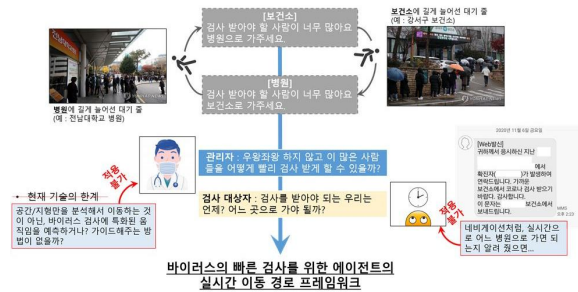


Fig. 1. Our framework for quick inspection of virus.

우리가 처해있는 COVID-19 상황에서는 이러한 기능을 해주는 컨트롤 시스템의 부재로 인해 상호작용이 제대로 이루어지지 않고, 결과적으로 선별진료소에는 대기 줄이 길어지고 (Fig 1 참조), 이런 환경에서 N차 감염뿐만 아니라, 의료봉사자들의 많은 희생을 요구하

게 되어 의료 붕괴까지 나타날 수 있는 상황이 올 수 있다.

현재 COVID-19 바이러스 검사를 받으려면 2가지 방법이 존재한다. 대표적으로 보건소와 병원을 가는 방법이지만, 사람들이 많이 있기 때문에 보건소에서는 병원으로, 병원에서는 보건소를 가라고 권장하고 있는 현실이다. 이러한 문제를 풀기 위해서는 아래와 같은 의문점에 대해서 답을 줄 수 있어야 한다. :

1. 바이러스 검사를 받아야 할 경우 어디로 가야 하는가? 문자로 가까운 병원/보건소 외에 좀 더 정확한 가이드 정보를 얻을 수는 없을까?
2. 현재 내 위치에서 가까운 병원은 어디지?
3. 현재 대기 시간은 얼마나 될까?
4. 임의의 위치에서 환자가 폭증할 경우, 그 환자들은 어느 병원을 가야 하는가? 각자 본인 동네 병원 가야 하는가?
5. 빠른 바이러스 검사를 하려면 어느 병원을 가야 할까?
6. 실시간으로 화자 이동 경로와 그로 인해 최적화된 이동 경로를 사람들에게 제공해줄 수 있다면 좋지 않을까?

본 논문에서는 바이러스의 확산과 N 차 감염을 최소화하기 위한 이동 경로 시스템을 제시한다. 사람들이 이동하는 데 있어서 그래프 기반 알고리즘을 활용하여 선별진료소 또는 일반 의료기관에 갈 수 있도록 안내하는 최적의 루트를 제공하고자 한다. 이 방법을 개발하는 데 있어서 실제 지도 API(예 : 구글 맵, 네이버 지도, 카카오 지도 등)를 이용하여 실제 적용이 가능하도록 디자인한다.

II. The Proposed Scheme

1. 빠른 바이러스 검사를 위한 도시 맵 기반 그래프 구성

사용자의 현재 위치에서 가장 근접한 n 개의 병원을 level로 표시한 l^0 , 해당 l^0 병원에 인접한 병원들을 l^1 , l^2 또는 l^3 까지, 레벨을 설정하여 병원 간 레벨이 겹치지 않고 그래프를 구성하는 방법이다. 이 방법은 그래프 알고리즘은 최소비용 최대-유량 알고리즘(Minimum Cost Maximum Flow)을 적용할 수 있는 방법으로 레벨이 겹치지 않게 환자를 보낼 수 있어 사람들의 접촉을 최소화하면서 이동을 하게끔 할 수 있다. 이동 수단의 개수가 제한이 있거나 이동에 어려움이 있는 상태일 경우, 긴급한 사고 발생 시 위와 같은 방법이 효율적으로 의료 산업에서 사용될 수 있다. 하지만 COVID-19 검사와 같이 감염률이 높은 상황에서는 접촉을 최소화하고 여러 병원을 거치기보다는 가까우면서 빠른 시간 내에 검사가 가능한 병원으로 안내하는 것이 가장 안정하고 효율적인 방법이기 때문에 위에서 언급한 방법은 빠른 바이러스 검사를 위한 좋은 그래프 구성 방법이 아니다.

따라서 본 논문에서는 레벨화 그래프의 노드 탐색 및 간선 연결 단계까지 동일하게 구성한 후, 레벨로 나뉜 그래프가 아닌, 찾아진 모든 병원에 환자 발생 지점으로부터 간선을 연결한다. 이때 가장 먼 병원의 거리를 범위로 설정하여 탐색해낸 병원들로 그래프를 구성한다 (Fig. 2 참조).

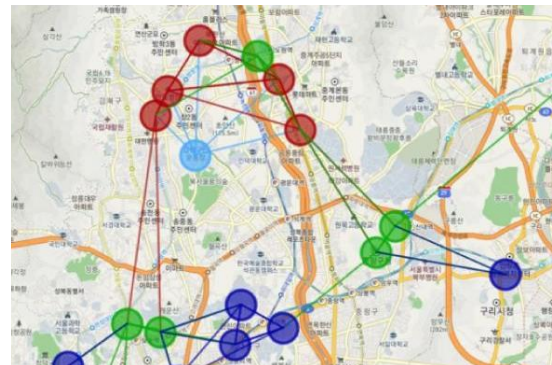


Fig. 2. Visualization of various level (sky blue : agent(user), red : l^1 , green : l^2 , blue : l^3).

본 논문에서는 지오메트리 프로세싱에서 사용하는 공간 분할 계층 알고리즘을 사용하여 사용자의 위치로부터 근접한 병원들을 빠르게 찾는다. 그래프를 구성과 근접할 병원을 찾을 때 NN(Nearest neighbor) 탐색 알고리즘 중 하나인 K -d 트리를 사용하여 최적화를 진행한다[1]. 병원 간의 거리는 2차원 거리가 아닌, 두 위경도 좌표 간의 거리이기 때문에 허버사인 공식(Haversine formula)을 이용하여 도시 맵에서의 거리를 계산한다 (수식 2 참조).

$$\begin{aligned} \Delta lon &= lon_2 - lon_1 \\ \Delta lat &= lat_2 - lat_1 \\ a &= \sin^2\left(\frac{\Delta lat}{2}\right) + \cos(lat_1)\cos(lat_2)\sin^2\left(\frac{\Delta lon}{2}\right) \\ c &= 2\arctan\left(\frac{\sqrt{a}}{\sqrt{1-a}}\right) \\ d &= Rc \end{aligned} \quad (2)$$

위 수식에서 R 은 지구의 반지름이며, lon 와 lat 는 각 좌표에 대한 위도와 경도이다.

이 값을 토대로 병원 위치기반의 K -d 트리와 그래프를 구축한다. 공간을 이산화시킬 수 있는 자료구조는 다양하지만, 실험적으로 K -d 트리가 가장 빠른 탐색 결과를 나타냈으며, 병원 간의 거리가 서로 다르므로 해시 테이블(Hash table)보다 안정적으로 그래프를 구축할 수 있었다. 또한, 사용자가 탐색 범위를 수정할 때도 인터랙티브하게 실행됨을 실험적으로 보여주었다. Fig. 3a는 병원 위치를 기반으로 남한 전체에 대한 K -d 트리 구성 결과를 보여주주고 있는 결과이고, Fig. 3b는 특정 지역에서 탐색한 주변 지역을 노드 형태로 보여주주고 있는 결과이다.

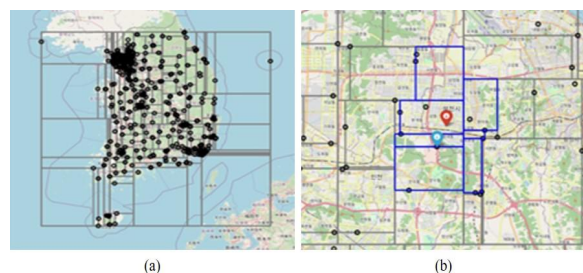


Fig. 3. K -d construction using hospital nodes.

2. 환자 이동을 위한 최적화된 비용 모델링

기존 경로 시스템의 경우에는 가까운 병원만을 알려주었으나 우리의 방법은 가까운 병원이 아닌, 이동 시간과 병원 대기 시간 및 규모까지 반영하여 최적의 안내를 할 수 있도록 설계한다. 본 논문에서 구성한 최종 그래프에서 환자들이 빠르고 효율적으로 이동할 수 있도록 병원의 규모, 현재 병원에 상주하고 있는 환자 수, 병원까지의 거리 등을 고려하여 최적의 비용을 계산한다. 비용이 낮을수록 최적화된 경로를 알려주며, 아래와 같은 수식을 이용한다 (수식 3 참조).

$$cost = t_{trva} + \frac{\max\left(P - S \frac{t_{trva}}{t_{treat}}, 1\right) + m}{S} t_{treat} \quad (3)$$

여기서 m 은 분배 인원이고, S 는 병원 규모, P 는 병원의 환자 수, t_{treat} 는 환자 한 명당 검사 시간, t_{trva} 는 병원까지 이동하는 시간을 나타낸다. 결과적으로 위 수식은 병원 규모가 클수록, 이동 시간이 짧을수록, 병원에 있는 환자 수가 적을수록 낮은 비용을 가지게 된다. 요약하면 다음과 같다. :

- 낮은 비용 : 대기 환자가 적고(▼), 거리가 가깝고(▼), 병원의 수용 가능 인원이 높은 상황(▲)
- 높은 비용 : 대기 환자가 많고(▲), 거리가 멀고(▲), 병원의 수용 가능 인원이 적은 상황(▼)

3. 환자 분배 방법

특정 병원에 환자가 몰리는 문제를 해결하기 위해 병원 간의 비용을 균일하게 분산되도록 매칭을 시켜야 한다. 이번 장에서는 균일한 비용을 유지할 수 있도록 환자를 분배하는 방법에 대해 설명한다.

환자 분배 규칙. 우리의 방법을 비용을 기반으로 환자들이 이동하기 때문에, 환자를 균일하게 이동시키려면 비용을 고르게 분산시켜야 할 필요성이 있다. 이를 위해 아래와 같은 순서로 비용을 갱신한다. :

1. 비용을 오름차순으로 정렬
2. 현재 비용인 C_1 이다음 비용인 C_2 와 최대한 비슷한 값이 될 때까지의 추가 이동 인원 m 을 계산
3. 현재 비용인 C_1 , 다음 비용인 C_2 , 그다음 비용인... C_n 와 같이 개수를 1개씩 증가하며 최종 인원(검사를 받아야 할 최종 환자 수)인 M 이 모두 분배될 때까지 반복

위와 같은 방법으로 환자를 분배했을 때, 특정 병원으로 몰리는 문제를 해결할 수 있었으며, Fig. 4는 환자 분배 전/후의 비용을 비교한 결과이다.

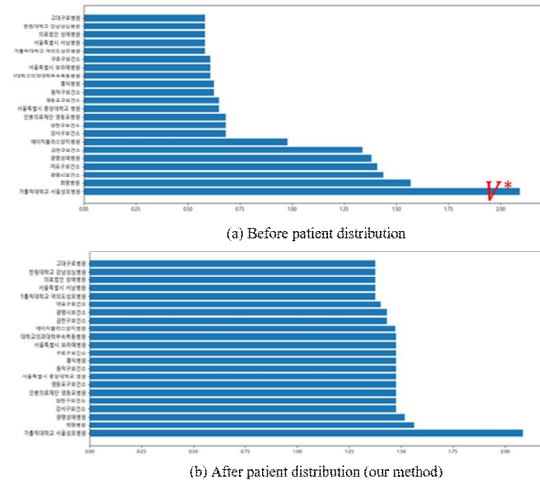


Fig. 4. Comparison of cost results before/after patient distribution.

Fig 4a에서 보듯이 환자를 분배하기 가까운 몇 개의 병원에 환자들이 몰리는 현상이 나타나는 것을 볼 수 있다. V^* 는 환자가 발생한 지역으로부터 가장 멀리 떨어진 병원이기 때문에 거기까지 환자가 도달한 가능성이 희박하다. 가장 먼 거리에 있는 병원까지의 이동 시간보다 주변 병원의 대기 시간이 더 짧아 사실상 환자를 받을 수 있는 상황이 거의 없으므로 상대적으로 비용이 크게 나온 것을 볼 수 있다. 하지만, 나머지 병원에서도 환자의 분배는 서로 다른 것을 볼 수 있다. 그에 비해 우리의 방법은 가장 멀리 떨어진 병원 외에서 거의 유사하게 비용이 분배된 것을 보여준다 (Fig 4b 참조). 비용 결과만 봤을 때도, 시간당 환자 처리에 대한 효율성을 잘 보여주고 있다.

접치는 경로 최적화. 새로운 환자들이 발생할 때마다 비용을 재계산해 경로를 갱신하면, 경로가 계속 변경되는 문제가 발생할 수 있다. 또한, 언제 어디서 새로운 환자가 발생할지 예상할 수 없으므로 병원이 접치는 새로운 환자가 발생했을 시, 먼저 이동 중인 환자가 치료받을 수 있도록 했다. 자세한 절차는 아래와 같다. :

1. 먼저 발생한 지역(R)에서 이미 이동하는 환자들에게 영향을 미치지 않도록 접치는 병원의 도착시간을 미리 계산
2. R 의 도착시간보다 나중에 발생한 지역(R_1)의 병원 도착시간이 빠를 경우 R 에서 계산된 비용이 변동되지 않는 최소한의 인원을 R_1 에 보냄
3. R 보다 R_1 이 병원 도착시간이 느릴 때는 병원에 R 에서 움직인 환자가 이미 도착해 있다고 가정하고 계산하여 인원을 보냄

Fig. 5는 중첩 경로 최적화를 적용했을 때 나타나는 결과이다. 각 병원의 위치에서 동 시간대에 환자들의 대기수를 시각화함으로써 중첩 경로 최적화에 대한 결과를 보여주고 있다. Fig 5b는 최적화를 적용하지 않은 결과이며, 상대적으로 대기수가 많다는 것을 원의 반지름 크기로 쉽게 알 수 있다. 반면에, 최적화를 적용한 우리의 방법은 같은 환경에서 환자 대기수가 훨씬 적다는 것을 반지름을

통해 보여주고 있으며, 이러한 결과를 향후 환자 이동을 시뮬레이션할 때 단위 시간당 더 많은 환자를 검사할 수 있다는 것을 나타낸다.

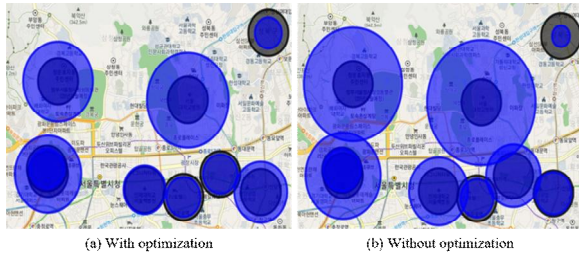


Fig. 5. Comparison of results with overlapping path optimization (circle : hospital location, radius of blue circle : number of patients waiting for treatment in hospital).

그래프의 부분 갱신 모든 병원을 대상으로 가까운 n 개의 병원을 찾으면 계산량이 너무 많아진다. 따라서 본 논문에서는 모든 병원에 대해 근접한 n 개의 병원을 연결한 그래프를 전처리 과정으로 구축하고, 이후 탐색 된 병원에서의 근접한 n 개의 병원을 찾을 때 전처리 과정에서 생성된 그래프를 사용함으로써 실행시간을 가속화한다. 이때 이전 시간에서 dt 시간이 지났다면, 그 병원의 근접한 n 개의 병원을 다시 탐색하여 재계산한다. dt 는 갱신 주기 시간으로써 사용자가 지정하거나 교통상황을 고려하여 유동적으로 변경되도록 설정하였다.

III. Results

본 논문에서 제안하는 방법의 효율성을 입증하기 위해 몇 가지 시나리오에 대해서 실험을 진행하였다. Fig. 6은 환자 발생 위치와 주변 병원 간의 관계를 고려하여 생성된 환자 이동 경로이다. 빨간색이 환자가 발생한 지역으로 주변 병원들로 빠르고 고르게 이동되는 결과를 보여주고 있다. Fig. 7은 실시간 교통상황을 고려한 환자 이동 경로이다. 실시간 교통상황을 적용하게 될 경우, 비용의 다양한 변수 중 병원까지 걸리는 시간에 변동이 생길 수 있다. Fig. 7a는 평일 낮(오후 1~2시) 와 비교적 차가 많이 막히는 시간대인 퇴근 시간(오후 6시)을 비교한 결과이다. 발생 시각 이외의 모든 상황이 동일함에도 불구하고, 도착 병원에 대한 목록이 변경된 것을 확인할 수 있다.

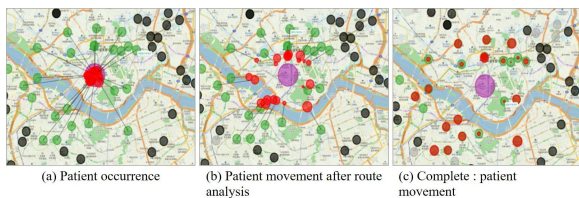


Fig. 6. Patient movement path according to the relationship between hospitals.

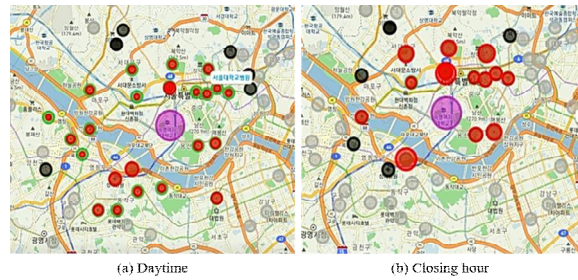


Fig. 7. Patient's movement route considering real-time traffic conditions.

IV. Conclusions

본 논문에서는 바이러스가 전파된 상황에서 빠른 검사를 위한 최적의 이동 경로 시스템을 제안하였다.

REFERENCES

[1] Zhou, Kun, et al. "Real-time kd-tree construction on graphics hardware." ACM Transactions on Graphics, Vol. 27, No. 5, pp. 1-11, 2008.