

Efficient Graph Construction and User Movement Path for Fast Inspection of Virus and Stable Management System

Jong-Hyun Kim*

*Professor, School of Software Application, Kangnam University, Yongin, Korea

[Abstract]

In this paper, we propose a graph-based user route control for rapidly conducting virus inspections in emergency situations (eg, COVID-19) and a framework that can simulate this on a city map. A* and navigation mesh data structures, which are widely used pathfinding algorithms in virtual environments, are effective when applied to CS(Computer science) problems that control Agents in virtual environments because they guide only a fixed static movement path. However, it is not enough to solve the problem by applying it to the real COVID-19 environment. In particular, there are many situations to consider, such as the actual road traffic situation, the size of the hospital, the number of patients moved, and the patient processing time, rather than using only a short distance to receive a fast virus inspection.

▶ **Key words:** Virus inspection, Pathfinding, Graph construction, Path trajectory, Urban-map

[요 약]

본 논문에서는 위급한 상황(예 : COVID-19)에서 바이러스 검사를 빠르게 진행하기 위한 그래프 기반 사용자 경로 제어와 이것을 도시 맵에서 시뮬레이션을 할 수 있는 프레임워크를 제안한다. 가상환경에서 많이 활용되는 길찾기(Pathfinding) 알고리즘인 A*나 네비게이션 메쉬 자료구조는 정해진 정적 이동 경로만을 안내하기 때문에 가상환경에서 에이전트를 제어하는 CS(Computer science)문제에 적용할 경우 효율적이다. 하지만, 실제 COVID-19 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 상황들이 많다. 본 논문에서는 위에서 언급한 다양한 속성들과 이를 이용한 최적화 함수를 모델링하여, 실제 도시 맵에서 바이러스 검사를 빠르고 효율적으로 처리할 수 있고, 다양한 상황을 디지털 트윈 방식으로 시뮬레이션을 할 수 있는 프레임워크를 제안한다.

▶ **주제어:** 바이러스 검사, 길찾기, 그래프 구축, 이동 경로, 도시 맵

-
- First Author: Jong-Hyun Kim, Corresponding Author: Jong-Hyun Kim
 - *Jong-Hyun Kim (jonghyunkim@kangnam.ac.kr), School of Software Application, Kangnam University
 - Received: 2022. 07. 08, Revised: 2022. 08. 18, Accepted: 2022. 08. 18.

I. Introduction

신종플루, 메르스, COVID-19처럼 바이러스는 인간과 동물을 지속적으로 위협하고 있으며, 현대화 사회에서 기술적인 진보가 있음에도 불구하고 바이러스는 여전히 위협적인 존재이다[1]. COVID-19의 확산 이후 3년이 지난 현재까지도 오미크론, 스텔스 등 여전히 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 참조).

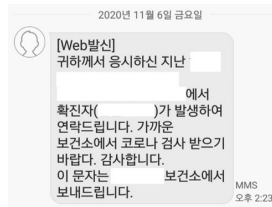


Fig. 1. Emergency text message.

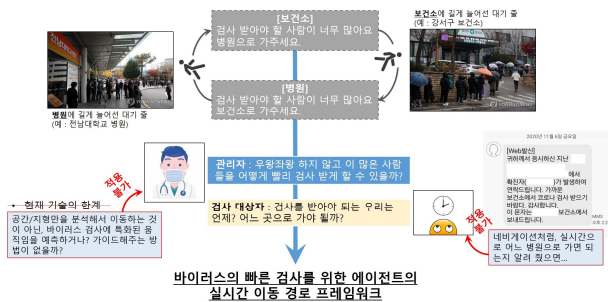


Fig. 2. Our algorithm overview for quick inspection of virus.

국내/외에 창궐해있는 COVID-19 상황에서도 이러한 기능을 제시해주는 컨트롤 시스템의 부재로 인해 관리자, 의사, 환자 등 서로간의 상호작용이 제대로 이루어지지 않고, 결과적으로 선별진료소에는 대기 줄만 길어지고 있는 상황이다 (Fig. 2 참조). 이런 환경에서는 N차 감염뿐만 아니라, 의료봉사자들에게 많은 희생을 요구하게 되어 의료 붕괴까지 나타날 수 있는 상황이 올 수 있다[2]. 현재 COVID-19와 같은 바이러스 검사를 받으려면 크게 2가지 방법이 존재한다. 대표적으로 보건소와 병원을 가는 방법이지만, 병원에서는 보건소를 가라고 권장하고, 보건소에서는 병원으로 가라고 서로 권장하고 있는 게 현실이다. 이러한 문제를 풀기 위해서는 아래와 같은 의문점에 대해서 답을 줄 수 있어야 한다.

- 1) 바이러스 검사를 받아야 할 경우 어디로 가야 하는가? 긴급재난문자로 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, 환자들은 어느 병원을 가야 하는가? 본인 동네 병원을 가야 하는가?
- 5) 빠른 바이러스 검사를 하려면 어느 병원을 가야 할까?
- 6) 실시간으로 환자 이동 경로와 최적화된 이동 경로를 사람들에게 제공해 줄 수 있다면 좋지 않을까?

본 논문에서는 바이러스의 확산과 N차 감염을 최소화하기 위한 이동 경로 시스템을 제안한다. 본 논문에서는 사람들이 이동하는 데 있어서 그래프 기반 알고리즘을 활용하여 선별진료소 또는 일반 의료기관에 이동할 수 있도록 안내하는 최적의 루트를 제시하고자 한다. 이 방법을 개발하는 데 있어서 실제 지도 API(예 : 구글맵, 네이버 지도, 카카오맵 등)를 이용하여 실제 적용이 가능하도록 설계한다.

문제정의. 최근 하드웨어의 발전에 따라 다수의 에이전트와 관련된 인터랙티브 응용프로그램이 발전을 하고 있다. 특히, MMORPG와 같은 게임과 가상현실 콘텐츠 같은 온라인 환경 혹은 메타버스에서 많은 에이전트들과 상호작용을 경험하게 된다[3,4]. 이러한 방식을 처리하는 접근법은 크게 3가지 방식이 존재하며 (에이전트 기반 움직임 제어 방법, 셀룰러 오토마타를 활용한 방법, 불연속적이고 연속적인 흐름을 이용한 방법), 자세한 방식은 아래와 같다 (Table 1 참조).

Table 1. A representative method of interacting with an agent in a virtual environment.

- 개별 에이전트의 로컬 정보를 기반으로 방향 및 규칙을 결정 : 효율적으로 무리를 짓기 위해 간단한 로컬 규칙을 사용한 연구가 있으며[5], 이 모델은 심리학이나 사회학적 선호도에 따른 움직임을 분석하는 분야로 확장됨[6]
- 장애물의 속도를 모션 규칙에 포함하여 충돌 회피를 개선한 방법이 있으며[7], 셀룰러 오토마타를 풀어냄으로써 에이전트 위치에 대한 진화 모델을 제안함[8].
- 흐름 기반 접근법은 다수 에이전트의 움직임을 물리 기반 유체 시뮬레이션에서 처리하는 방식과 유사하게 계산하는 방법 : 2차원 공간에서 에이전트를 유체 입자들로 취급함[9]
- 연속적인 알갱이 흐름과 연속체 역학을 기반으로 모션을 제어하는 연구도 있음[10]

게임이나 가상환경에서 가장 많이 활용되는 A*탐색 알고리즘은 인터랙티브 환경에서 실시간 경로 찾기를 효율적으로 계산할 수 있다[11] (Table 2 참조).

Table 2. Pathfinding process using A* search algorithm.

- A*알고리즘에 비해 D*는 부분적으로만 알려진 환경이나 변화하는 동적 환경에서 검색 정확도가 높은 방법임[12]
- 사각형 격자, 쿼드트리, 컨벡스 폴리곤 및 네비게이션 메쉬 등을 이용하여 환경을 양자화하는 방식이 있음[13]
- 네비게이션 메쉬는 에이전트가 넓은 가상공간에서 길을 쉽게 찾는데 도움을 주기 위해 사용되는 자료 구조이며, 메쉬의 형태는 일반적으로 그래프 구조로 구현되고, 이러한 구조는 다양한 알고리즘으로 개선됨[14]

최근에 가상환경이나 로봇공학에서 많이 사용되는 네비게이션 메쉬 접근법을 제안하는 방법에 활용할 수 있을까? 이 대답을 위해 우선 기본적인 네비게이션 메쉬가 동작하는 절차에 대한 이해가 필요하다 (Table 3 참조).

Table 3. Pathfinding process using Navigation mesh algorithm.

- 네비게이션 메쉬를 기반으로 길찾기를 하는 절차는 일반적으로 프로세싱과 런타임 단계로 구성됨
- 프로세싱 단계
 - (1) 지형 데이터를 로드하고, 네비게이션 메쉬 생성 등이 이루어짐
 - (2) 로드된 지형 데이터를 사용하여 에이전트 이동이 허용되는 부분이 네비게이션 메쉬로 처리됨
 - (3) 이러한 이동성 영역을 식별할 때는 지형의 기술기 정보를 이용하거나 개발자의 수작업으로 식별 입력이 진행됨
- 런타임 단계
 - (1) 목적지를 설정하고 최단 거리 탐색과 에이전트 이동 과정이 계산됨
 - (2) 에이전트의 이동은 이벤트 트리거 또는 사용자 인터랙션에 따라 지정되며, 최단 거리 탐색 단계에서 A*알고리즘이 적용되어 네비게이션 메쉬 내부의 대상 지점까지 최단 경로를 계산
 - (3) 이동 단계에서는 계산된 최단 경로를 따라 에이전트가 이동되며, 생성된 네비게이션 메쉬 내부의 로컬 편차로 인해 다른 에이전트와의 충돌을 회피하면서 이동함

위에서 설명한 네비게이션 메쉬에 대한 이벤트 트리거로 환자 발생을 인지할 수는 있지만, 지형만을 고려하기 때문에 본 논문에서 제안하는 방법과는 차이가 있으며, 그 외에도 가장 큰 차이는 다음과 같다.

- 1) 지형이 아닌, 사용자의 현 위치에서 이동하는 실제 거리와 도로 교통을 고려한 최단거리를 설계
- 2) 사용자뿐만 아니라, 사용자-병원 간의 관계성을 고려하여 최단거리 알고리즘을 설계
- 3) 임의의 위치에서 환자들이 추가 발생한 경우에도 최적의 경로를 제안할 수 있어야 함
- 4) 사용자가 원하는 경로뿐만 아니라, 컨트롤타워에서 실시간으로 환자의 수용 및 대기 수, 그리고 갑작스럽게 발생한 추가 환자의 이동경로까지 확인할 수 있어야 함

II. Preliminaries

1. Related work

개별적인 에이전트의 움직임. 비디오 게임에서 인공지능은 플레이어에게 몰입을 줄 수 있게 하는 중요한 요소이다. 그 중 하나는 실제 사람의 행동과 유사하게 가상환경 내 에이전트의 움직임을 만드는 것이다[15]. 길찾기는 비디오 게임과 같은 가상환경에서 흔히 요구되고 적용되는 기본 기술이다[16]. 이런 기술은 한 지점에서 다른 지점으로의 최적경로를 찾는 지도 분석에 활발하게 응용되고 있다. 가장 짧은 경로, 복잡하지 않은 교통 방식, 가장 안정한 교통 방식과 같은 몇 가지 기준과 점수에 따라 최적의 경로가 결정된다[17].

길찾기 알고리즘은 비디오 게임에서 많은 연구가 진행되었다[16]. A*알고리즘과 그 변형 기법들은 이미 많은 가상환경에서 에이전트의 움직임을 제어하는데 활용되고 있다[18,19]. 앞에서 언급했듯이 이 방법은 길찾기 문제를 해결하는데 충분히 좋은 방법이지만, 환경이 정적이기 때문에 다양한 분야에 적용하기에는 한계가 있다. 동적 환경은 환경이 변할 때마다 정적인 것으로 가정하므로 높은 계산 시간을 필요로 한다.

다양한 문제에서 정확도와 호환성을 향상시키기 위해 A* 경로 찾기를 활용한 추가 연구도 많이 수행되었다[20]. 대표적으로 IDA*(Iterative deepening A*)는 A*알고리즘에서 요구하는 메모리를 절약할 수 있는 방법을 제시했다[21]. HPA*(Hierarchical pathfinding A*)는 가상환경을 클러스터라는 직사각형 블록으로 변환하는 맵 처리 방법이며, A*보다 빠른 계산이 가능하다[22]. A*경로 찾기를

가속화하기 위한 또 다른 연구에는 HAA*(Hierarchical annotated A*)[23]와 PRA*(Partial refinement A*)[24] 기법들도 있다. 이 외에 Block A*는 맵 데이터베이스를 이용하여 비디오 게임에서 장애물을 회피할 수 있는 방법을 제시했다[25].

실시간 경로 찾기에는 대표적으로 TBA*(Time-bound A*)[26]가 있다. 이 방법은 탐색 환경이 넓으면 시간이 오래 걸리는 단점이 있다. 또한, KNN LRTA*(K nearest neighbor LRTA*) 방법은 에이전트가 목표지에 더 가깝다고 보장하는 방향으로 하위 목표치를 계산한다[27]. 이 방법에서 에이전트는 맵뿐만 아니라 서브 격자에서도 움직인다. 그 후에 이들은 HCDPS(Hill-climbing and dynamic programming search)[28]방법을 추가하여 좀 더 세밀한 계산이 가능한 방법을 제안했다. 실시간 길찾기인 HCDPS 방법은 여전히 정적 환경에서만 실행되는 한계가 있다. A*와 유전(Genetic) 알고리즘을 결합한 RTP-GA(Real-time pathfinding with Genetic algorithm)[29] 방법은 장애물 없는 길찾기(Barrier-free path) 방식을 이용하여 에이전트가 목표를 좀 더 효율적으로 찾아가는데 활용했다. 다른 방법에 비해 RTP-GA의 장점은 동적 환경에서 궤적을 계산할 수 있는 유전자 알고리즘을 사용한다는 점과 이 알고리즘으로 선택된 최적의 경로는 A*를 활용 할 수 있다.

무리 짓는 집단행동. 가상환경에서 무리 짓는 행동은 개체들이 서로 움직이고 상호작용 할 수 있는 능력을 가진 집단이다. 무리 짓는 행동에는 세 가지 주요 규칙이 있다 [30] : 1) 다른 개체들로부터 무리 지역 중심에 가까이 머무르는 힘인 응집력, 2) 다른 개체와 멀리 떨어져 충돌의 가능성을 피하는 충돌회피, 3) 다른 개체들의 속도와 유사한 속도를 가지려는 힘인 속도 일치. 또 다른 접근법인 인공전위장(Artificial potential field, APF)은 에이전트를 제어하는 장을 전처리 과정에서 계산하는 방법이다. APF는 경로 계획 로봇 공학(Path planning robotics)에서 Khatib에 의해 처음으로 적용되었다[31]. APF는 개체들을 서로 끌어당기는 Attractive potential field와 반발력인 Repulsive potential field로 구성된다[32].

III. The Proposed Scheme

본 논문에서는 사용자의 현재 위치에서 가장 근접한 n 개의 병원을 레벨로 표시한 l^0 , 해당 l^0 병원에 인접한 병원들을 l^1, l^2 또는 l^3 까지 레벨을 설정하여 병원 간 레벨을 겹치지 않게 그래프를 구축하는 방법이다. 이 방법은 그래프 알고리즘인 최소-비용 최대-유량 알고리즘

(Minimum-cost maximum flow)을 적용하여 풀 수 있는 문제이다 : 레벨이 겹치지 않게 환자를 보낼 수 있어 사람들의 접촉을 최소화하면서 이동을 할 수 있다. 본 논문에서 제안하는 방법은 이동 수단의 개수에 제한이 있거나 이동에 어려움이 있는 경우 혹은 긴급한 사고 발생 시 위와 같은 방법이 효율적으로 의료 산업에서 활용될 수 있다. 하지만, COVID-19 검사와 같이 감염률이 높은 상황에서 접촉을 최소화하고 여러 병원을 거치기보다는 가까우면서도 빠른 시간 내에 검사가 가능한 병원으로 안내하는 것이 가장 안전하고 효율적인 방법이기 때문에 위에서 언급한 방법은 빠른 바이라스 검사를 위한 좋은 그래프 구성 방법이 아니며, 이 문제를 해결할 수 있는 방식을 소개한다.

1. City map based graph construction for quick virus scans

본 논문에서는 레벨화 그래프의 노트 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. 3 참조).

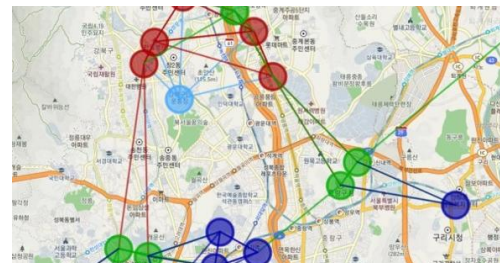


Fig. 3. Visualization of node level (sky blue : agent(patient), red : l^1 , green : l^2 , blue : l^3).

본 논문에서는 지오메트리 프로세싱에서 활용되는 공간 분할 계층 알고리즘을 사용하여 사용자 위치로부터 근접한 병원들을 빠르게 찾는다. 그래프 구성과 인접한 병원들을 찾을 때는 최단거리 탐색 알고리즘 중 하나인 K-d 트리를 사용하여 최적화한다. 병원 간의 거리는 2차원 거리가 아닌, 위도와 경도 좌표 간의 거리이기 때문이 허버사인 공식(Haversine formular)을 이용하여 도시 맵에서의 거리를 계산한다 (수식 1 참조).

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

위 수식에서 R 은 지구의 반지름이며, lon 와 lat 는 각 좌표에 대한 위도와 경도이다. 이 값을 토대로 병원 위치 기반 K -d 트리와 그래프를 구축한다. 공간을 이산화하는 자료구조는 다양하지만 실험적으로 K -d 트리가 가장 빠른 검색 결과를 나타냈으며, 병원 간의 거리가 서로 다르므로 해시 테이블(Hash table)보다 안정적으로 그래프를 구축할 수 있었다. Fig. 4a는 병원의 위치를 기반으로 남은 전체에 대한 K -d 트리 구성 결과를 보여주고 있고, Fig. 4b는 특정 지역에서 탐색한 주변 지역을 노드 형태로 보여주고 있는 결과이다.

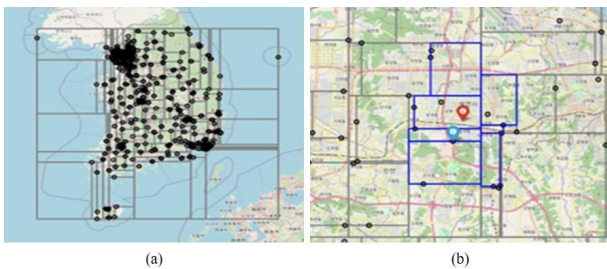


Fig. 4. Construction of K -d tree on hospital nodes.

2. Optimized cost modeling for patient movement

앞에서 언급한 기존의 경로 시스템은 가까운 병원만을 알려주었으나 본 논문에서 제안하는 방법은 가까운 병원이 아닌, 이동 시간과 병원 대기 시간 및 규모까지 반영하여 최적의 이동 안내를 할 수 있도록 한다. 다시 말해, 본 논문에서 제안하는 최종 그래프를 통해 환자들이 빠르고 효율적으로 이동할 수 있도록 병원의 규모, 현재 병원에 상주하고 있는 환자의 수, 병원까지의 거리 등을 고려하여 최적의 비용을 계산한다. 비용이 낮을수록 최적화된 경로임을 나타내며, 아래와 같은 수식을 이용한다 (수식 2 참조).

$$C = t_{trav} + \frac{\max\left(P - S \frac{t_{trav}}{t_{treat}}, 1\right) + m}{S} t_{treat} \quad (2)$$

여기서 m 은 분배 인원이고, s 는 병원의 규모, P 는 병원의 환자 수, t_{treat} 는 환자 한명 당 검사 시간, t_{trav} 는 병원까지 이동하는 시간을 나타낸다. 결과적으로 위 수식은 병원 규모가 클수록, 병원까지의 이동 시간이 짧을수록, 병원에 있는 환자의 수가 적을수록 낮은 비용을 가지게 된다. 이를 요약하면 아래와 같다 (Table 4 참조).

Table 4. Examples of situations based on cost.

- 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. Patient distribution

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. 본 논문에서 제안하는 방법은 앞에서 언급한 비용 기반으로 환자들이 이동하기 때문에 환자를 균일하게 이동시키려면 비용을 고르게 분산시켜야 한다. 이를 위해 아래와 같은 순서로 비용을 갱신한다 (Table 5 참조).

Table 5. Updating cost.

1. 비용을 오름차순으로 정렬
2. 현재 비용인 C_i 이, 다음 비용인 C_j 와 최대한 비슷한 값이 될 때까지 추가 이동 인원 m 을 계산
3. 현재 비용인 C_i , 다음 비용인 C_j , 그 다음 비용인,...., C_n 와 같이 개수를 1씩 증가하며 최종 인원 (검사를 받아야 할 최종 환자 수)인 M 이 모두 분배될 때 까지 반복

위와 같은 방법으로 환자를 분배했을 때, 특정 병원으로 몰리는 문제를 해결할 수 있었으며, Fig. 5는 환자 분배 전과 후의 비용을 비교한 결과이다.

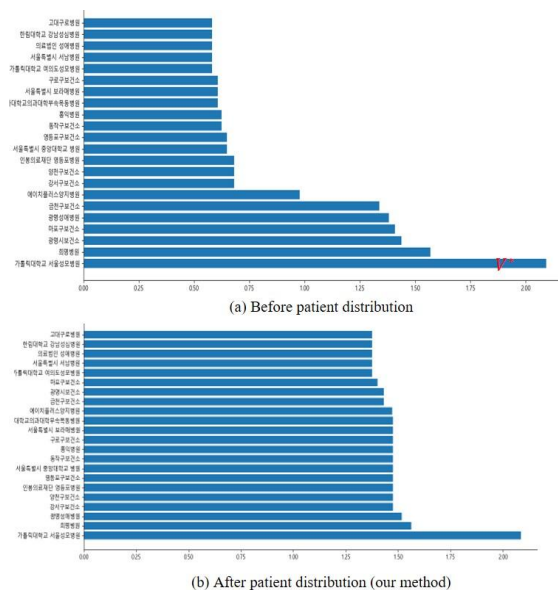


Fig. 5. Comparison of cost results before/after patient distribution.

Fig. 5a에서 보듯이 환자를 분배하기 가까운 몇 개의 병원에 환자들이 몰리는 현상이 나타나는 것을 볼 수 있다. 여기서 v^* 는 환자가 발생하는 지역으로부터 가장 멀리 떨어진 병원이기 때문에 거기까지 환자가 도달할 가능성이 희박하다. 가장 먼 거리에 있는 병원까지의 이동 시간보다 주변 병원의 대기 시간이 더 짧아지기 때문에 사실상 환자를 받을 수 있는 상황이 거의 없으므로 상대적으로 비용이 크게 나온 것을 볼 수 있다. 하지만, 나머지 병원들에서도 환자의 분배는 서로 다른 것을 볼 수 있다. 그에 비해 본 논문에서 제안하는 방법은 가장 멀리 떨어진 병원 외에서 거의 유사하게 비용이 분배된 것을 보여준다 (Fig. 5b 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 때, 본 논문에서는 먼저 이동 중인 환자가 치료를 받을 수 있도록 했다. 이를 위한 절차는 아래와 같다 (Table 6 참조).

Table 6. Path optimization process.

| |
|---|
| <ol style="list-style-type: none"> 1. 먼저 발생한 지역(R)에서 이미 이동하는 환자들에게 영향을 미치지 않도록 접치는 병원의 도착시간을 미리 계산 2. R의 도착시간마다 나중에 발생 지역(R_1)의 병원 도착시간이 빠를 경우 R에서 계산된 비용이 변동되지 않는 최소한의 인원을 R_1에 보냄 3. R보다 R_1이 병원 도착시간이 느릴 때는 병원에 R에서 움직인 환자가 이미 도착해 있다고 가정하고 계산하여 인원을 보냄 |
|---|

Fig. 6은 중첩 경로 최적화를 적용했을 때 나타나는 결과이다. 각 병원의 위치에서 동 시간대에 환자들의 대기수를 시각화함으로써 중첩 경로 최적화에 대한 결과를 보여주고 있다. Fig. 6b는 중첩 경로 최적화를 적용하지 않은 결과이며, 상대적으로 대기수가 많다는 것을 원의 반지름 크기로 쉽게 알 수 있다. 반면에 최적화를 적용한 본 논문의 방법은 같은 환경에서 환자 대기수가 훨씬 적다는 것을 반지름을 통해 보여주고 있으며 (Fig. 6a 참조), 이러한 결과는 환자 이동 시뮬레이션을 할 때 단위 시간당 더 많은 환자를 검사할 수 있다는 것을 나타낸다.

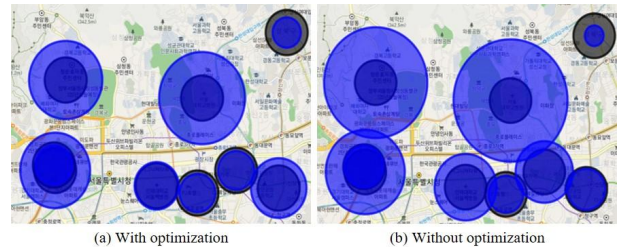


Fig. 6. Comparison of results with overlapping path optimization (circle : hospital position, radius of blue circle : number of patients waiting for treatment in hospital).

그래프의 부분 갱신. 모든 병원을 대상으로 가까운 n 개 병원을 찾으면 계산이 너무 많아진다. 따라서 본 논문에서는 모든 병원에 대해 근접한 병원들을 연결한 그래프를 전처리 과정으로 구축하고, 이후 환자 발생 장소에서 인접한 n 개의 병원을 찾을 때 전처리 과정에서 생성된 로컬 그래프를 사용함으로써 실행시간을 단축시킨다. 이때 이전 시간에서 dt 시간이 지났다면, 그 병원에서 근접한 n 개의 병원을 다시 탐색하여 재계산한다. 여기서 dt 는 갱신 주기 시간으로써 사용자가 지정하거나 교통상황을 고려하여 유동적으로 변경되도록 설정한다.

IV. Results

본 연구의 결과들을 만들기 위해 실험한 환경은 Intel Core i7-7700K CPU, 32GB RAM, Geforce GTX 1080Ti GPU가 탑재된 컴퓨터를 이용했다. 본 연구에서 제안하는 방법의 효율성을 입증하기 위해 몇 가지 시나리오에 대해서 실험을 진행했다. 실험 데이터는 카카오맵 API를 이용했으며, 상대적으로 교통이 혼잡한 수도권과 퇴근시간을 기준으로 시뮬레이션을 함으로써 이 구간에서 환자들의 이동이 효율적으로 제어되는지를 확인하고자 한다.

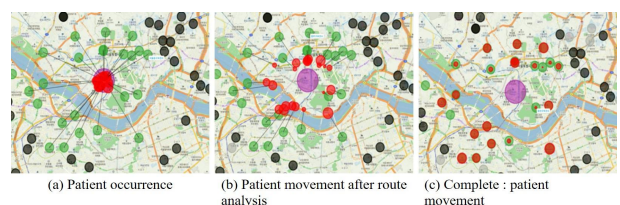


Fig. 7. Patient movement path according to the relationship between hospitals.

Fig. 7은 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. 7a에서는 빨간색 구의 위치에서 환자가 발생했음을 보여주는 이벤트이며, 이로 인해 주변 병원들로 환자들이 이동하는 과정을 구의 반지름을 통해 보여주고 있다. 환자들이 처음 발생했을 때 큰 구의 크기가 주변 병원들로 흡수되어 환자들이 치료를 받음에 따라 작은 크기의 구로 변화는 모습을 잘 보여주고 있다 (Fig. 7b와 7c 참조).

Fig. 8은 실시간 교통상황을 고려한 환자의 이동 경로이다. 실시간 교통상황을 적용하게 될 경우, 다양한 변수 중 병원까지 걸리는 시간인 t_{travel} 가 변동될 수 있다. Fig. 8a는 평일 낮(오후 1~2시)과 비교적 차가 많이 막히는 시간대인 퇴근시간(오후 6시)을 비교한 결과이다. 발생 시각 이외의 모든 상황이 동일함에도 불구하고, 도착 병원에 대한 목록이 변경된 것을 확인할 수 있다.

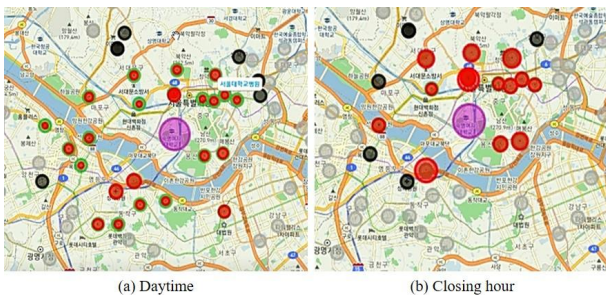


Fig. 8. Patient's movement rate considering real-time traffic conditions.

V. Conclusions

본 논문에서는 실제 지도 데이터와 $K-d$ 트리를 이용하여 빠르게 지도 내 병원 데이터에 접근할 수 있는 트리 구성기법을 제안했다. 또한, 환자 이동 경로를 최적화하기 위한 새로운 비용 함수와 그에 따른 환자 이동 기법, 그래프 갱신 기법, 경로 중첩 최적화 등을 제안했다. 에이전트를 이동시키기 위한 다양한 길찾기 알고리즘을 통해 일반적인 환자 이동은 가능하지만, 병원과 환자간의 그리고 도로 교통간의 상황을 고려한 이동 시스템은 본 논문에서 제안하는 방법이 최초이며, 현재 세계적으로 위협을 가하는 COVID-19 상황에서 더욱더 효율적으로 활용될 수 있을 거라 기대한다.

그럼에도 불구하고 몇 가지 한계점이 있다. 본 논문에서 제안한 이동경로 최적화를 위한 주체는 일반인도 될 수가 있다. 제안하는 방법이 환자, 감염병의심자, 그리고 일반인까지 분류해서 경로 최적화를 처리할 수 있는 방법은 아

니다. 단지, 이 기능이 필요할 것 같다고 판단하는 본인 스스로가 되거나, 혹은 관리자 입장에서 다수의 인원을 디지털트윈 방식으로 시뮬레이션 해 볼 수 있다는 점에서 의미가 있다. 하지만, 환자 감염병의심자, 그리고 일반인을 분류하여, 서로간의 접촉이 최소화될 수 있도록 경로 최적화를 해야 될 것으로 보인다. 향후 환자의 이동만이 아닌, 치료목적이나 잠재적인 환자의 이동을 예측할 수 있는 프레임워크로 확장할 계획이다.

REFERENCES

- [1] Pujari, Rishita, Mary V. Thommana, Brisandi Ruiz Mercedes, and Ayna Serwat. "Therapeutic options for COVID-19: a review." *Cureus* 12, no. 9 (2020).
- [2] Choi, Jun Yong. "COVID-19 in South Korea." *Postgraduate medical journal* 96, no. 1137 (2020): 399-402.
- [3] Lee, Bong-Keun, Choong-Shik Park, Jae-Hong Kim, Sang-Jo Youk, and Keun Ho Ryu. "An intelligent npc framework for context awareness in mmorpg." In *2008 International Conference on Convergence and Hybrid Information Technology*, pp. 190-195. IEEE, 2008.
- [4] Assiotis, Marios, and Velin Tzanov. "A distributed architecture for MMORPG." In *Proceedings of 5th ACM SIGCOMM workshop on Network and system support for games*, pp. 4-es. 2006.
- [5] Ashida, Koji, Seung-Joo Lee, Jan M. Allbeck, Harold Sun, Norman I. Badler, and Dimitris Metaxas. "Pedestrians: Creating agent behaviors through statistical analysis of observation data." In *Proceedings Computer Animation 2001. Fourteenth Conference on Computer Animation (Cat. No. 01TH8596)*, pp. 84-92. IEEE, 2001.
- [6] Pelechano Gómez, Núria, Kevin O'Brien, Barry G. Silverman, and Norman Badler. "Crowd simulation incorporating agent psychological models, roles and communication." In *First International Workshop on Crowd Simulation*. 2005.
- [7] Van Den Berg, Jur, Sachin Patil, Jason Sewall, Dinesh Manocha, and Ming Lin. "Interactive navigation of multiple agents in crowded environments." In *Proceedings of the 2008 symposium on Interactive 3D graphics and games*, pp. 139-147. 2008.
- [8] Dadova, J. "Cellular automata approach for crowd simulation." Faculty of Mathematics, Physics and Informatics, Comenius University (2012).
- [9] Helbing, D. "Self-Organized pedestrian Crowd Dynamics and Design Solutions Transportation Science." (2003).
- [10] Cristiani, Emiliano, Benedetto Piccoli, and Andrea Tosin. "Multiscale modeling of granular flows with application to crowd dynamics." *Multiscale Modeling & Simulation* 9, no. 1

- (2011): 155-182.
- [11] Hart, Peter E., Nils J. Nilsson, and Bertram Raphael. "A formal basis for the heuristic determination of minimum cost paths." *IEEE transactions on Systems Science and Cybernetics* 4, no. 2 (1968): 100-107.
- [12] Stentz, Anthony. *Optimal and efficient path planning for unknown and dynamic environments*. CARNEGIE-MELLON UNIV PITTSBURGH PA ROBOTICS INST, 1993.
- [13] Stout, Bryan. "The basics of A* for path planning." *Game Programming Gems 1* (2000): 254-263.
- [14] Guzman, E. "Valve thrills fps fans with new counter-strike game." *2D-X-The Best Damn Video Game Site Period* (2011).
- [15] Millington, Ian, and John Funge. *Artificial intelligence for games*. CRC Press, 2018.
- [16] Botea, Adi, Bruno Bouzy, Michael Buro, Christian Bauckhage, and Dana Nau. "Pathfinding in games." *Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik*, 2013.
- [17] Yap, Peter. "Grid-based path-finding." In *Conference of the canadian society for computational studies of intelligence*, pp. 44-55. Springer, Berlin, Heidelberg, 2002.
- [18] Hart, Peter E., Nils J. Nilsson, and Bertram Raphael. "A formal basis for the heuristic determination of minimum cost paths." *IEEE transactions on Systems Science and Cybernetics* 4, no. 2 (1968): 100-107.
- [19] Cowling, Peter I., Michael Buro, Michal Bida, Adi Botea, Bruno Bouzy, Martin V. Butz, Philip Hingston, Héctor Muñoz-Avila, Dana Nau, and Moshe Sipper. "Search in real-time video games." *Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik*, 2013.
- [20] Cowling, Peter I., Michael Buro, Michal Bida, Adi Botea, Bruno Bouzy, Martin V. Butz, Philip Hingston, Héctor Muñoz-Avila, Dana Nau, and Moshe Sipper. "Search in real-time video games." *Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik*, 2013.
- [21] Korf, Richard E. "Depth-first iterative-deepening: An optimal admissible tree search." *Artificial intelligence* 27, no. 1 (1985): 97-109.
- [22] Pohl, Ira. "Heuristic search viewed as path finding in a graph." *Artificial intelligence* 1, no. 3-4 (1970): 193-204.
- [23] Harabor, Daniel, and Adi Botea. "Hierarchical path planning for multi-size agents in heterogeneous environments." In *2008 IEEE Symposium On Computational Intelligence and Games*, pp. 258-265. IEEE, 2008.
- [24] Sturtevant, Nathan, and Michael Buro. "Partial pathfinding using map abstraction and refinement." In *AAAI*, vol. 5, pp. 1392-1397. 2005.
- [25] Yap, Peter, Neil Burch, Robert Craig Holte, and Jonathan Schaeffer. "Block A*: Database-driven search with applications in any-angle path-planning." In *Twenty-Fifth AAAI Conference on Artificial Intelligence*. 2011.
- [26] Björnsson, Yngvi, Vadim Bulitko, and Nathan R. Sturtevant. "TBA*: Time-Bounded A*." In *IJCAI*, pp. 431-436. 2009.
- [27] Bulitko, Vadim, and Yngvi Björnsson. "kNN LRTA*: Simple subgoaling for real-time search." In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, vol. 5, no. 1, pp. 2-7. 2009.
- [28] Bulitko, Vadim, Yngvi Björnsson, and Ramon Lawrence. "Case-based subgoaling in real-time heuristic search for video game pathfinding." *Journal of Artificial Intelligence Research* 39 (2010): 269-300.
- [29] Machado, Alex Fernandes da V., Ulysses O. Santos, Higo Vale, Rubens Gonçalves, Tiago Neves, Luiz Satoru Ochi, and Esteban W. Gonzalez Clua. "Real time pathfinding with genetic algorithm." In *2011 Brazilian Symposium on Games and Digital Entertainment*, pp. 215-221. IEEE, 2011.
- [30] Reynolds, Craig W. "Steering behaviors for autonomous characters." In *Game developers conference*, vol. 1999, pp. 763-782. 1999.
- [31] Khatib, Oussama. "Real-time obstacle avoidance for manipulators and mobile robots." In *Autonomous robot vehicles*, pp. 396-404. Springer, New York, NY, 1986.
- [32] Zhou, Li, and Wei Li. "Adaptive artificial potential field approach for obstacle avoidance path planning." In *2014 Seventh International Symposium on Computational Intelligence and Design*, vol. 2, pp. 429-432. IEEE, 2014.

Authors



Jong-Hyun Kim received the B.A. degree in the Department of Digital Contents at Sejong University in 2008. He received M.S. and Ph.D. degrees in the Department of Computer Science and Engineering at Korea University,

in 2010 and 2016. He is an Associate Professor in the School of Software Application in Kangnam University. His current research interests include physics-based simulation and phenomenon modeling and virtual production.