

# 부드러운 헤비사이드 계단 함수 기반의 NavMesh 제어 기법을 이용한 효율적인 NPC의 경로 추적

김중현<sup>○</sup>, 김수균<sup>\*</sup>

<sup>○</sup>강남대학교 소프트웨어응용학부,

<sup>\*</sup>제주대학교 컴퓨터공학과

e-mail: jonghyunkim@kangnam.ac.kr

## Efficient Path Tracking of Non-Player Character with Controlling NavMesh Based on Smoothed Heaviside Step Function

Jong-Hyun Kim<sup>○</sup>, Soo Kyun Kim<sup>\*</sup>

<sup>○</sup>School of Software Application, Kangnam University,

<sup>\*</sup>Dept. of Computer Eng., Jeju National University

### ● 요약 ●

본 논문에서는 사용자의 다양한 물리적 속성 중 부드러운 헤비사이드 계단 함수와 다양한 물리적 속성(속도, 시점 등)을 활용하여 가중치 맵을 계산하고 이로부터 논플레이어 캐릭터(Non-player character, NPC)의 경로를 효율적으로 제어할 수 있는 NavMesh 제어 기법을 제시한다. 게임과 같은 가상환경에서 NPC는 일반적으로 네비게이션 메쉬(Navigation mesh, NavMesh)를 이용하여 이동한다. 하지만, NavMesh는 정적인 형태이기 때문에 사용자에게 의해 디자인되어야 하고, 이러한 문제를 완화하고자 자동으로 NavMesh를 업데이트하는 기술이 연구되고 있지만, 메쉬 복원을 자동화할 뿐 실제 NPC 행동 제어라고 하기에는 힘든 접근법이다. 본 논문에서는 동적 네비게이션 프레임워크를 유지한 채, 사용자의 시점과 물리적 특성을 통해 NPC를 효율적이고 정확하게 경로 제어할 수 있는 방법을 제안하고, NavMesh의 형태에만 의존하던 NPC의 움직임을 완화하여 좀 더 사실적인 경로 제어를 보여준다.

**키워드:** 경로 제어(Controlling path), 경로 찾기(Pathfinding), 사용자의 시점(User's viewpoint), 부드러운 헤비사이드 계단 함수(Smoothed Heaviside step function), 게임(Game), 가상환경(Virtual environment)

### I. Introduction

가상환경에서 NPC의 움직임은 사용자와 상호작용을 하는 개체로서 매우 중요한 역할을 한다. NPC의 움직임이 부자연스러우면 사용자 입장에서는 몰입도가 저해될 수 있기 때문에 반드시 고려되어야 할 부분이다. 사용자인 PC(Player character)의 움직임은 정해진 형태가 아니고, 매번 변경되고 예측하기 어렵기 때문에 상황마다 NPC의 움직임을 제어하는 것 또한 풀기 어려운 문제이다. 최근에 쿼드트리와 경로 유사도 기법을 이용하여 이 문제를 완화시키려는 기법들이 제안되었다[1,2]. 하지만, 이 방법은 NPC가 이동할 때 가이드를 해주는 NavMesh를 자동으로 업데이트하는 문제에 초점이 맞춰져 있으며, PC와의 상호작용에 의한 경로 제어라고는 할 수 없다.

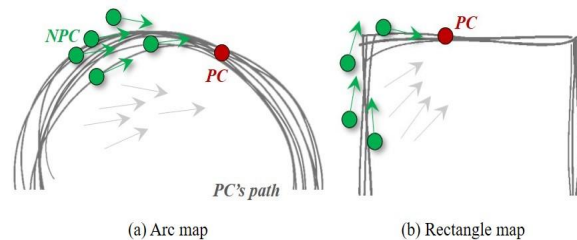


Fig. 1. Previous pathfinding issues (gray line : updated shortest path by PC, gray arrow : NPC's path with PC's position and viewpoint).

이 접근법들은 NavMesh의 업데이트를 자동으로 해줌으로써 다자 이니의 수작업을 줄여줄 수 있다. 물론, 이러한 문제는 인공지능을 통해 풀 수도 있지만, 학습을 위한 데이터 구축 및 정제가 어려운 문제이다. 결과적으로 사용자의 메타 데이터를 매번 정제하는 것은 어려운 접근법이기 때문에 이전 기법들에서는 공통적으로 사용자의 경로 데이터를 통해 NavMesh를 구축하여 사용한다. Fig. 1에서 보듯이 PC의 경로를 이용하는 경우 누적 데이터에 따라 회색라인처럼 NavMesh가 구축된다. 초기에는 경로 정확도가 떨어지지만, 데이터가 누적됨에 따라 사용자가 주로 가는 경로는 기반으로 NavMesh가 업데이트된다. 본 논문에서는 사용자의 위치만으로 구축된 NavMesh가 아닌, 시점 및 속도를 통해 좀 더 도적인 NavMesh를 구축하고 업데이트할 수 있는 방법을 제시한다.

## II. The Proposed Scheme

본 논문에서는 클라이언트 서버에 저장되어 있는 사용자의 궤적 데이터를 대신, Kim 등의 방법을 통해 사용자의 움직임과 유사한 경로 데이터를 얻는다. 이 데이터를 통해 이동 궤적에 따른 밀도 맵과 경로 유사도 방식을 이용하여 기본적인 동적 NavMesh를 생성한다 (Fig. 1 참조).

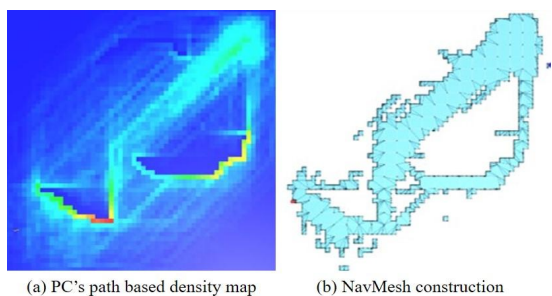


Fig. 2. Visualization of density map and NavMesh.

NavMesh만을 이용하게 된다면, Fig. 2b에서 보이는 삼각형 메쉬 위로만 NPC들이 움직이기 때문에 거리가 멀어지면 쫓아기는 일반적인 행동을 취할 수 없게 된다. 거리가 멀어지면 빠르게 쫓아가려는 행동 대신, 정해진 길목을 통해서만 움직이려는 문제가 나타나며, 이러한 특징은 Fig. 1에서도 보여주고 있다. 우리는 이 문제를 해결하기 위해 헤비사이드 계단 함수를 통해 NavMesh를 제어하는 기법을 소개한다.

NPC는 2가지 상태인  $P^n$ 와  $P^c$ 로 움직임을 제어된다. 여기서  $P^n$ 는 NavMesh로 인해 움직이는 경로이며,  $P^c$ 는 PC를 향해 움직이는 시점을 기반으로 제어된 움직임이다. 헤비사이드 계단 함수에 의해서 제어된 NPC의 움직임은 아래와 같다 (수식 1 참조).

$$P^*(x) = (1 - \delta(w))P^n(x) + \delta(w)P^c(x) \quad (1)$$

$$\delta(w) = \begin{cases} 0 & w \leq -\epsilon, \\ \frac{1}{2} - \frac{w}{2\epsilon} - \frac{1}{2\pi} \sin\left(\frac{\pi w}{\epsilon}\right) & -\epsilon \leq w \leq \epsilon, \\ 0 & w \geq \epsilon. \end{cases} \quad (2)$$

여기서  $\delta(w)$ 는 부드러운 헤비사이드 계단함수이며, 이 값을 기반으로 NavMesh의 영향력이 달라진다 (Fig. 3 참조). 여기서 distance는 NavMesh의 뼈대화(Skeletonization)의 포인트 위치로부터 NPC 사이의 거리를 나타낸다. 결과적으로 가중치가 작으면 NavMesh의 형태에 따라 NPC가 움직이게 되고, 거리의 차가 클수록 가중치가 크게 되어 PC의 시점과 방향에 따라 NPC의 움직임에 자동으로 제어되는 결과를 가지고 온다.

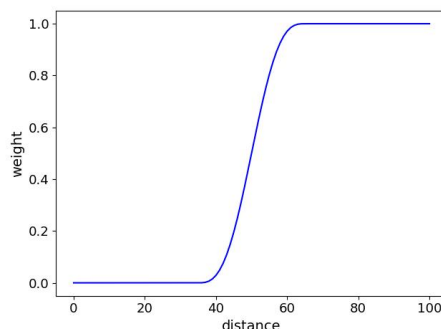


Fig. 3. The graph of weight parameter  $\delta(w)$  with Equation 2.

## III. Conclusions

본 논문에서는 정적인 NavMesh에만 의존하는 NPC의 움직임을 좀 더 효율적으로 제어할 수 있는 부드러운 헤비사이드 계단 함수 기반 NavMesh 제어 기법을 제안했다.

## REFERENCES

- [1] Shin-Jin Kang, YongO Kim and Chang Hun Kim, "Live path: adaptive agent navigation in the interactive virtual world," The Visual Computer, Volume 25, Number 5-7, pp. 469-477, 2010.
- [2] Kim, Jong-Hyun, Jung Lee, and Sun-Jeong Kim. "Navigating non-playable characters based on user trajectories with accumulation map and path similarity." Symmetry, Volume 12, Number 10, pp. 1592, 2020.