JKSCI

한국컴퓨터정보학회논문지
Journal of The Korea Society of Computer and Information
Vol. 25 No. 4, pp. 55-61, April 2020
https://doi.org/10.9708/jksci.2020.25.04.055

Efficient Controlling Trajectory of NPC with Accumulation Map based on Path of User and NavMesh in Unity3D

Jong-Hyun Kim*

*Professor, Dept. of Software Application, Kangnam University, Yongin, Korea

[Abstract]

In this paper, we present a novel approach to efficiently control the location of NPC(Non-playable characters) in the interactive virtual world such as game, virtual reality. To control the NPC's movement path, we first calculate the main trajectory based on the user's path, and then move the NPC based on the weight map. Our method constructs automatically a navigation mesh that provides new paths for NPC by referencing the user trajectories. Our method enables adaptive changes to the virtual world over time and provides user-preferred path weights for smartagent path planning. We have tested the usefulness of our algorithm with several example scenarios from interactive worlds such as video games, virtual reality. In practice, our framework can be applied easily to any type of navigation in an interactive world.

▸ **Key words**: Controlling trajectory of NPC, User's path, Physical properties, Pathfinding, Unity3D

[요 약]

본 논문에서는 사용자의 경로를 기반으로 가중치 맵을 계산하고, 이를 이용하여 게임이나 가상현실과 같은 인터랙티브 가상 환경에서 논플레이어 캐릭터(Non-playable characters, NPC)의 위치를 효율적으로 제어할 수 있는 방법을 제시한다. 우리의 방법은 사용자의 움직임을 참조하여 NPC에 새로운 경로를 제공하는 네비게이션 메쉬를 자동으로 구성한다. 우리의 방법은 시간이 지남에 따라 사용자가 주로 다니는 길목을 정확하게 찾아내기 때문에 가상환경에 적응형으로 자동 변경이 가능하고, 사용자가 선호하는 경로를 가중치로 스마트 에이전트와 같은 움직임을 만들어 낼 수 있다. 우리는 비디오 게임이나 가상현실과 같은 인터랙티브 환경의 몇 가지 예제 시나리오를 통해 본 논문에서 제안하는 알고리즘의 유용성 테스트를 실험했다. 실제로 우리의 프레임워크는 인터랙티브 환경을 활용하는 모든 유형의 탐색에 쉽게 적용 할 수 있다.

▸ **주제어**: 논플레이어 캐릭터의 경로 제어, 사용자의 경로, 물리적 속성, 경로 찾기, 유니티3D

• First Author: Jong-Hyun Kim, Corresponding Author: Jong-Hyun Kim
*Jong-Hyun Kim (jonghyunkim@kangnam.ac.kr), Dept. of Software Application, Kangnam University
• Received: 2020. 03. 16, Revised: 2020. 04. 01, Accepted: 2020. 04. 01.

http://www.ksci.re.kr pISSN:1598-849X | eISSN:2383-9945

I. Introduction

실제 환경에서 사람이 가보지 않은 지역에는 도로나 길목이 없다. 그러나 시간이 지남에 따라 사람들이 이러한 영역을 탐색하고 경로로 표시하기 시작함으로써 점차 길목으로 변화되어 사람이 다닐 수 있는 도로가 생기게 된다. 다른 사람들은 먼저 온 사람들에 의해 만들어진 길을 자연스럽게 따르고, 결과적으로 길은 점차 확장되고 길목으로써 더욱더 명확해 진다. 실제 환경에서의 이러한 자연적인 길목 생성 과정에서 영감을 받아 본 논문에서는 인터랙티브 가상 환경을 위한 NPC의 새로운 탐색 방식을 소개한다. 제안하는 시스템은 인터랙티브 환경에서 특징적인 사용자 움직임을 샘플링 한 후, 이전에 없었던 새로운 경로를 생성한다. 새롭게 생성된 경로를 사용하여 이전 정적 세계를 확장하거나 기존 NPC 경로로 찾지 못했던 부적합한 행동을 완화할 수 있다.

(a) (b)

Fig. 1. Inappropriate behavior of NPC in games (red dotted line :wrong movement)

정적인 네비게이션 메쉬에 따라 NPC들이 움직이다가 메쉬가 없는 영역을 마주치게 되면 목적지를 앞에 두고 우회하거나, Fig. 1에서 보듯이 때때로 NPC이 잘못된 행동으로 벽에 끼는 문제가 발생한다. 이러한 상황은 사용자의 움직임을 고려하지 못한 네비게이션 메쉬 문제이며, 결과적으로 NPC의 움직임이 전혀 다른 방향으로 움직이게 되어 구석진 곳에서 빠져나오지 못하는 또 다른 문제로 이어지기도 한다. 이러한 문제는 게임의 몰입도를 저하시킬 뿐만 아니라, 게임의 수익과도 직접적으로 연결되는 구조이기 때문에 매우 중요한 문제이다.

최근 그래픽 하드웨어가 발전함에 따라 다수개의 NPC와 관련된 인터랙티브 응용 프로그램이 날로 발전하고 있다. 결과적으로 사용자는 비디오 게임이나, 가상현실 콘텐츠에서 많은 수의 NPC들가 상호작용할 수 있고, MMORPG(Massively Multiplayer Online Role-Playing Games)와 가상현실과 같은 온라인 환경에서 더 많은 상호작용을 경험하게 된다. 이러한 환경에서 실시간 경로 찾기 접근법은 사전에 정의된 환경에서 NPC에 대한 최적의 경로를 실시간으로 선택할 수 있는 프로세스이며, 전통적으로 컴퓨터 그래픽 및 로봇 분야에서 연구되어 왔다.

많은 실시간 경로 찾기 접근법은 공간 복잡성을 줄이기 위해 로드맵 또는 그래프 기반 방법으로 경로를 글로벌하게 최적화시키는 방법에 중점을 두었다. Kavraki 등은 정적 세계에서 확률적 로드맵을 사용하여 그래프 기반 경로 찾기 기술을 제시했다 [1]. 이들은 전처리 단계에서 주어진 환경을 통해 로봇의 움직임에 대한 로드맵을 미리 내장시키고, 특정 경로를 쿼리해야 되는 경우 빠른 그래프 검색을 사용하여 로드맵에서 경로를 검색한다. 이러한 방법은 다양한 컴퓨터 애니메이션 문제들을 풀어내는데 있어서 성공적으로 적용되었다 [2,3]. 이러한 확률적 로드맵 접근 방식은 차원이 높은 환경에서 적합하지만, 실시간 프로세스에서는 경로 생성이 저품질로 나타나기 때문에 적합하지 않다.

경로 생성의 성능과 품질을 동시에 만족시키기 위해 Two-level 경로 계획 방법이 제안 되었다 [4,5]. 이 방법에서 첫 번째 Level은 목표를 향한 글로벌 경로를 계산하고, 두 번째 Level에서 로컬 충돌 회피 및 탐색을 계산한다. 전역 경로는 NPC가 없는 장면에서 정적 객체를 나타내는 로드맵 그래프를 사용하여 계산한 뒤, 검색 알고리즘을 사용하여 목표 위치에서 그래프 노드까지의 거리를 계산한다.

가상 세계의 크기와 복잡성이 증가함에 따라 매우 큰 가상 환경에서 실시간 경로 탐색을 위한 솔루션도 많은 분야에서 요구되고 있다. Pettr's 등은 큰 가상 환경인 군중 시뮬레이션에서 위 문제를 다루었다 [6]. 이들은 네비게이션 그래프를 생성하는 공간 구조화 기술을 군중 시뮬레이션에 활용하여 복잡한 실시간 경로 탐색 문제를 해결하였다. Pettr's 등은 이 기법을 예측 접근 방식으로 확장하여 보행자 간의 상호작용을 고려한 자율 네비게이션 방법을 제안했다 [7]. 이 방법은 확장 가능한 시뮬레이션 루프를 지원하기 때문에 계산 리소스를 공간과 시간에 분산시키면서 군중 상황을 업데이트 시킬 수 있고, 결과적으로 대규모 가상 세계에서 경로 찾기를 효율적으로 적용할 수 있다. 최근 가상 세계는 역동적인 환경으로 변화하고 있다. Sud 등은 동적인 장면에서 다중 NPC의 경로 탐색에 대한 새로운 접근법을 소개했다 [8]. 그들은 보로노이 다이어그램을 활용하여 계산된 MaNG(Multi-agent Navigation Graph)를 사용했다. 또한, 이들은 입자 기반 시뮬레이션을 사용하는 반응형 로드맵 그래프인 AERO(Adaptive Elastic Roadmaps)를 제안했다 [9].

이전 로드맵 또는 탐색 그래픽 기반 경로 찾기 기술은 환경 자체의 데이터를 이용하여 가상 세계에서 NPC 탐색에

중점을 두었다. 가상현실에서의 NPC의 움직임을 사실적으로 제어하려는 연구 분야는 현실성에 대한 사용자의 요구를 충족시키기 위해 몇 년 만에 기하급수적으로 증가했다. 가상 세계에서 사용자는 제한 없이 특정 NPC를 제어하고 다른 NPC와 상호작용이 가능하다. 그러나 인터랙티브 환경의 불규칙성과 복잡성 때문에 이전의 경로 탐색 기법만 사용하여 이러한 유형의 환경을 처리하기에는 한계가 있다. 우리의 방법은 이러한 유형의 길 찾기 문제를 해결하기 위해 제안되었다. 본 논문에서는 사용자 상호작용 데이터에 기반 한 새로운 적응형 네비게이션 방법을 제안한다.

II. Preliminaries

1. Related works

A^*알고리즘은 인터랙티브 환경에서 실시간 경로 찾기에 가장 널리 활용되는 검색 방법이다 [10]. D^*(Dynamic A^* algorithm)는 부분적으로만 알려진 환경이나 변화하는 환경에 대응하고자 A^*알고리즘에 비해 더욱더 검색 정확도가 높은 방법이다 [11]. A^*기반 알고리즘에서는 환경의 특성에 따라 정적으로 양자화된 검색 공간(Quantized research space)이 필요하다. Stout 등은 사각형 격자, 쿼드 트리, 볼록 다각형 및 네비게이션 메쉬 등을 이용하여 환경을 양자화하는 방법을 제안했다 [12]. 네비게이션 메쉬는 에이전트가 넓은 가상공간에서 길을 쉽게 찾는데 도움을 주기위해 사용되는 데이터 구조이다. 메쉬의 형태는 일반적으로 그래프 구조로 구현되며, 다양한 분야에서 확장 개발되고 있다. 네비게이션 메쉬의 가장 일반적인 용도 중 하나는 비디오 게임 [13,14] 및 상업용 길 찾기 미들웨어이다 [15,16].

네비게이션 메쉬를 기반으로 길 찾기를 하는 절차에는 일반적으로 전처리 단계와 런타임 단계의 두 단계가 있다. 전처리 단계에서는 1) 지형 데이터를 로드, 2) 네비게이션 메쉬 생성 등이 이루어진다. 로드된 지형 데이터를 사용하여 실제 에이전트 이동이 허용되는 영역에 네비게이션 메쉬로 처리된다. 이러한 영역을 식별할 때는 지형의 기울기 정보를 이용하거나 개발자의 수작업으로 식별이 진행된다.

런타임 단계에서는 1) 목적지 설정, 2) 최단 경로 검색, 3) 이동 등이 이루어진다. 에이전트의 이동은 이벤트 트리거 또는 사용자 인터랙션에 따라 지정되며, 최단 경로 검색 단계에서는 A^*알고리즘이 적용되어 대상 지점까지 최단 경로를 계산한다. 이동 단계에서는 계산된 최단 경로를 따라 이동하고 생성된 네비게이션 메쉬 내부의 로컬 편차로 인해 다른 에이전트와의 충돌을 피하면서 이동한다.

인터랙티브 가상 환경에서 에이전트는 PC(Playable characters)와 NPC(Non-playable characters)로 분류된다. NPC의 움직임은 PC에 비해 단순하며, 위에서 언급한 최단 경로 탐색 과정에서 얻은 경로를 따라 이동한다. 최단 경로 탐색 단계에서 획득한 네비게이션 메쉬로 커버되지 않는 영역은 NPC가 엑세스 할 수 없는 영역이 된다. PC의 이동 과정은 인터랙티브 환경에서의 NPC와는 약간 차이가 있다. PC 이동은 포인트 지정 또는 방향 제어에 의해 제어된다. 여기서, 포인트 지정은 사용자가 PC의 목적지를 직접 지정하는 것을 말하며 일반적으로 마우스 클릭을 통해 이루어지고, 방향 제어는 현재 위치에서 PC의 이동 방향에 대한 사용자 입력을 말한다. 포인팅으로 지정된 대상이 있는 PC는 NPC와 동일한 방식으로 네비게이션 메쉬를 사용하여 경로를 찾고, 방향키를 사용하여 대상을 정의한 PC는 탐색 메시를 참조하지 않고 지정된 각 방향으로 사전 설정된 거리만큼 이동한다. 최근에 Kopel와 Hajas는 인공지능을 통해 비디오 게임에서 NPC 에이전트의 행동을 결정하는 방법을 제안했다 [17]. 이들은 인공지능 기법을 통해 에이전트의 움직임을 제어하려 했지만 알고리즘이 복잡할 뿐만 아니라, 네트워크를 통해 수렴될 수 있는 신빙성있는 메타데이터를 만드는 일이 어렵기 때문에 산업에서 활용되기 어려운 구조이다.

III. The Proposed Scheme

본 논문에서는 인터랙티브 환경에서 나타나는 경로 찾기 방법에 대한 문제를 해결하기 위해 사용자의 이동궤적 데이터를 이용하여 새롭게 에이전트의 경로를 찾는 기법을 제안한다. 핵심 아이디어는 사용자가 캐릭터를 제어하는 경우 본능적으로 더 나은 경로를 선택할 수 있다는 가정이며, 사용자가 캐릭터를 제어할 때는 이동궤적을 고려하여 실제 사람이 길을 찾는 방식과 유사하게 제작한다. 게임이나 가상현실과 같이 초기 환경에서는 사용자의 이동궤적 정보가 없기 때문에 최적의 NPC 경로를 찾는 것이 어렵다. 하지만, 시간이 지날수록 사용자는 환경에 익숙해지고, 그 만큼 신뢰 있는 데이터가 쌓이면서 더 나은 길을 찾기 시작한다. 예를 들어, 예기치 않게 PC가 점프를 하거나 길이 없는 곳을 지정해서 갈 수 있지만, NPC는 이러한 예기치 않은 행동을 할 수 없다. 만약에 PC의 경험으로 최적화된 경로를 학습하고 이러한 결과를 가상 환경 내 경로 찾기에 통합할 수 있다면 사용자에게 더 몰입감을 줄 수 있을 것이다. 이러한 과정을 물리적 속성을 기반으로 한

경로 확장이라고 하며, 본 논문에서는 특정 알고리즘이 아니라 실제 통계 데이터를 기반으로 설계한다.

인터랙티브 환경에서 PC는 자신의 플레이어 경험을 통해 실력이 향상될 수 있지만, 일반적으로 NPC는 할 수 없기 때문에 시간이 지남에 따라 PC는 똑똑해지고 NPC는 초기 정적 동작을 유지하게 된다. 이런 문제로 인해 사용자는 게임을 하면서 지루해 하거나, 이런 에이전트의 취약점을 악용하기도 한다. 진화 알고리즘이나 인공지능 신경망이 이 문제에 대한 해결책이 될 수도 있지만, NPC의 움직임을 올바르게 제어하기 위해서는 다양한 매개 변수에 기반 한 적절한 목적 함수가 필요하다. 사용자의 다양한 물리적 속성 중에 움직임에 영향을 주는 위치를 이용하여 NPC의 움직임을 개선하는데 사용하며, 이 데이터는 본 논문에서 NPC를 발전시키는데 사용되는 필수 메타데이터이다. NPC는 사용자의 이전 물리적 속성을 알고 있기 때문에, 보다 현명한 결정을 내릴 수 있게 된다.

이러한 기능을 구현하기 위해 우리는 사용자의 경로 데이터를 임의로 제작한다. 시작지점과 목적지가 결정되면 실제 사용자의 움직임과 유사한 행동패턴을 계산하여 데이터를 수집하고, 이것을 이용하여 NPC의 경로를 진화시키고, 결과적으로 경로가 제어되는 결과를 보여준다.

1. Data collection of user's path

에이전트의 진화를 위한 메타데이터를 수집하기 위해 아래와 같은 수식을 이용하여 불규칙적인 움직임인 p^*를 계산한다 (수식 1 참조).

$$p^*(x) = (1-\delta)Q(x) + \delta R(x) \quad (1)$$

$$\delta = \frac{\| p_{str} - p_{des} \|}{\| p_{cur} - p_{des} \|} \quad (2)$$

여기서 $p_{str}, p_{cur}, p_{des}, p^*$는 출발, 현재, 도착위치를 각각 나타내며, Q는 A^*알고리즘과 같이 최단 경로를 통해 얻어진 위치이고 R은 관찰을 하면서 움직이는 패턴을 모델링하기 위한 랜덤 지수다 (2차원에서 4가지 방향을 갖는다), 일반적으로 사람은 출발지 부근에서는 어디로 가면 좋을지 관찰하는 행동패턴으로 지그재그한 움직임이 나타나는 반면, 목적지에 가까울수록 명확하게 찾아간다. 이러한 움직임 기반으로 모델링한 가중치가 δ이다 (수식 2 참조). 출발위치에 가까울수록 δ가 증가되기 때문에 불규칙한 움직임이 강하게 표현되고, 도착위치에 가까울수록 δ가 감소하여 불규칙한 움직임이 거의 표현되지 않고 명확하게 목적지를 향해 움직이게 된다.

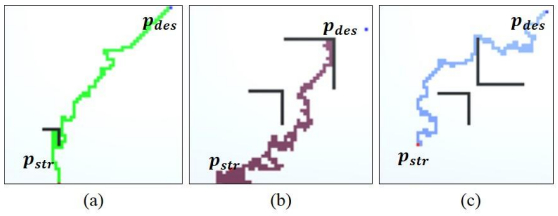


Fig. 2. NPC's path using Equation 1

Fig. 2에서 보듯이, p_{str}에서 p_{des}로 진행하는 과정에서 R로 인해 불규칙하게 탐색하는 과정이 표현된다. 하지만, 장애물이 오목한 경우 NPC들이 장애물을 빠져나가지 못하고 갇히는 경우가 발생하고, 이 문제를 피하기 위해 본 논문에서는 4가지 조건에서 실험을 진행한다.

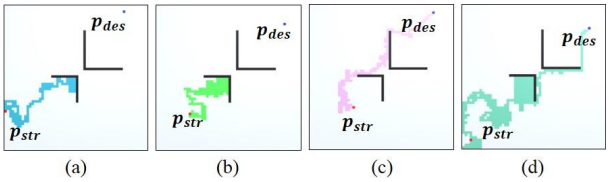


Fig. 3. Visualizing path of NPC with various conditions

*Cond.*1은 지나갔던 길을 다시 못 지나가게 설정하는 것이다. 이 같은 조건은 반복적으로 같은 영역을 맴도는 문제를 피하기 위함이지만, 결과적으로 NPC의 불규칙한 패턴인 R이 포함되어 있기 때문에 랜덤하게 맴돌다가 사방에 막혀 죽는 상황이 발생한다 (Fig. 3a 참조). 특히, 장애물이 시작위치에 근접해 있다면 δ가 커지게 되므로 랜덤 패턴에 의해 NPC의 움직임이 더 불안정하게 된다. *Cond.*2는 지나간 길을 갈 수 있도록 허용하는 것이며, 이 같은 조건은 결과적으로 도착지점으로부터 가까운 구석자리를 계속적으로 맴도는 움직임을 보였으며, 특히 이 문제는 오목한 장애물 형태에서 더욱더 심각하게 나타났다 (Fig. 3b 참조). *Cond.*3은 누적치 맵을 이용하여 같은 장소를 3번만 지나갈 수 있게 설정하고, 그 이상이 되면 그 곳은 다시 못가도록 막는 방법이다. 앞에서 소개한 2개의 조건보다는 빠져나갈 확률이 높았지만, 대체적으로 불규칙한 패턴을 고려하지 못하거나, 주변을 맴돌다가 제대로 된 경로를 찾지 못하는 상황이 발생한다 (Fig. 3c 참조). 마지막으로, *Cond.*4는 R과 누적치 맵을 같이 사용하는 것이다. 우리는 4가지 조건들에 대해서 매번 p_{str}, p_{des}, 장애물을 랜덤하게 설정한 뒤 1,000번의 실험을 진행했다. 각 조건들에 대한 성공 확률이 *Cond.*1 : 10%, *Cond.*2 : 13%, *Cond.*3 : 20%, *Cond.*4 : 96%와 같은 결과가 나왔다. 최종적으로는 *Cond.*4를 이용하여 메타데이터를 수집하였다 (Fig. 3d 참조).

2. Generation of accumulation map with user's path

사용자가 자주 가는 경로를 찾기 위해 1,000개의 움직임 데이터를 이용했으며, 히스토그램 형식으로 지나간 경로를 누적하여 밀도 맵 η를 구축한다.

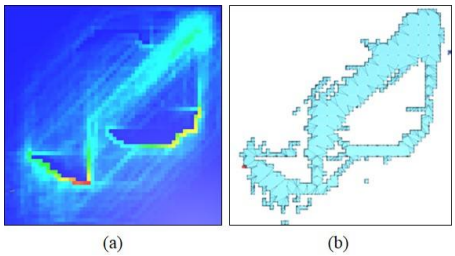


Fig. 4. Visualizing accumulation map(a) and refined navigation mesh(b)

Fig. 4a는 이동궤적에 따른 누적 맵이며, 빨간색에 가까울수록 사용자가 많이 지나간 영역이다. 이 맵을 이용하여 NPC의 위치를 제어하기 위해 Unity3D에서 제공하는 NavMesh를 이용하여 네비게이션 메쉬를 생성한다 (Fig. 4b 참조). 사용자가 주로 다니던 길만을 추출하기 위해 누적 맵의 밀도 값이 0.1 이상인 영역에 대해서만 네비게이션 메쉬인 κ를 생성한다. 앞에서 설명한 과정으로 κ 생성까지 성공했다면, 이 메쉬를 이용하여 NPC를 움직일 수 있다 (Fig. 5 참조). κ는 η의 임계값이 의존하여 메쉬를 생성하기 때문에 임계값이 너무 작거나 크면 κ가 불안정하게 생성되며 (Fig. 5b 참조), 결과적으로 경로가 끊기는 현상이 발생하기 때문에 NPC가 제대로 움직이지 못한다.

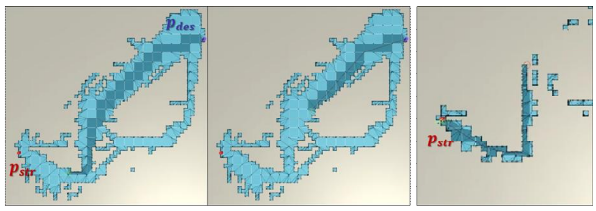


Fig. 5. Visualizing path of NPC with η and unstable case : (a) NPC's path (threshold of η : 0.1), (b) Unstable navigation mesh

임계값에 따라 민감하게 끊기지 않고 사용자가 자주 가는 경로를 기반으로 NPC의 위치를 제어하기 위해 PC의 경로와 κ사이의 경로-유사도(Path-similarity)에 따라 유효한 PC 경로를 수집한다. 이 과정에서 유사성은 PC의 경로와 κ의 좌표들이 많이 겹친다면 유사성이 높다고 판단했으며, 유사도가 70% 이상 되는 PC의 경로를 최종적으로 수집한다. Fig. 6a와 b는 유효한 경로로 판단되기 때문에 실제로 네비게이션 메쉬를 생성할 때 사용되며, Fig 6c는 κ와 유사도가 많이 떨어지기 때문에 경로생성에 영향을 주지 않는다.

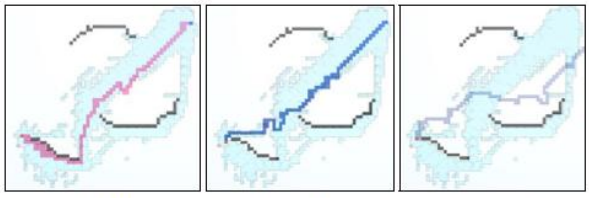


Fig. 6. Valid and invalid paths of PC with path-similarity

Fig. 7은 유사도와 경로 개수가 네비게이션 메쉬에 어떠한 영향을 끼치는지를 보여주는 그림이다. Fig. 7의 결과들은 Fig. 5b와 비교했을 때 끊어지는 영역 없이 안정적으로 경로를 생성했다. 유사도와 수집한 경로의 개수에 따라 길이 갈라지는 분기점에 대해서는 서로 다른 형태를 나타냈지만 경로의 큰 방향성에 대해서는 차이가 크지 않고, 끊기지 않게 경로를 생성했다 (Fig. 7 참조).

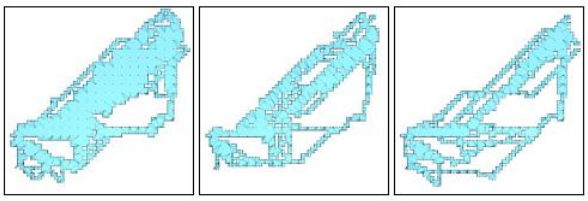
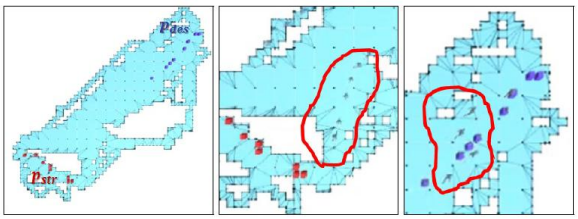


Fig. 7. Visualizing path of NPC with path-similarity (% : similarity, num : number of collected path data)

IV. Results

본 연구의 결과들을 만들기 위해 실험한 환경은 Intel Core i7-7700K CPU, 32GB RAM, Geforce GTX 1080Ti GPU가 탑재된 컴퓨터를 이용하였다. 우리는 본 논문에서 제안하는 방법의 우수성을 판단하기 위해 다양한 환경에서 NPC들의 움직임을 실험했으며, 대부분 안정적인 움직임을 생성했고, 정적인 NPC의 움직임이 아닌 사용자의 움직임을 고려한 결과들을 만들어 냈다.



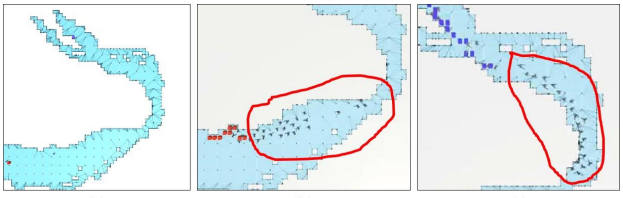
(a) Entire navigation mesh (b) Frame : 21 (c) Frame : 65

Fig. 8. Movement of NPCs with navigation mesh (red circle : p_{str}, blue circle : p_{des}, red region : NPCs)

Fig. 8은 대각선 방향으로 시작과 도착위치를 설정하고, NPC의 움직임을 실험한 결과이다. 여기서 시작점과 도착지점은 한 곳에서 몰리는 움직임을 피하기 위해 다 수개의 지점으로 설정했으며 빨간색은 시작위치, 파란색은 도착위치를 의미한다. 결과적으로 NPC들은 사용자의 경로 기반으로 생성된 네비게이션 메쉬를 활용하여 도착지점까지 순조롭게 이동한다. Fig 8a와 b에서 빨간색으로 테두리친 영역이 NPC들이 이동하고 있는 모습이다.

Fig. 9는 곡선 형태로 움직이는 PC들을 기반으로 네비게이션 메쉬가 생성되고, 이에 따라 NPC들이 이동되는 결과를 보여주는 장면이다. Fig. 8의 결과처럼 직선 형태 뿐만 아니라 곡선 형태의 움직임에서도 안정적으로 결과를 만들어 냈다.

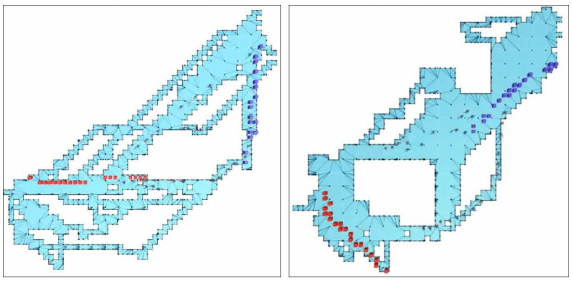
Fig. 10은 다양한 장면에서 실험해본 결과이다. 앞에서 언급했듯이 제안하는 방법은 η의 임계값에 의존하여 분기점의 개수나 형태가 달라질 수 있지만 길이 끊어지지 않고 시작위치에서 도착지점까지 안정적으로 NPC들이 이동되는 결과를 잘 보여주었다.



(a) (b) (c)

Fig. 9. Curved movement of NPC

지금까지의 결과들에서 보여주었듯이, NPC의 이동경로는 PC의 경로에 의존하기 때문에 동일한 맵에서도 항상 다른 형태로 네비게이션 메쉬가 생성 될 수 있다. 이러한 의미는 동일한 맵에서 항상 유사한 움직임을 보였던 기존 NPC의 한계점을 해결 할 수 있다는 것을 의미한다. 게임과 같은 가상현실 콘텐츠는 수많은 맵을 사용하며, 특히 각 맵에서 유효한 영역을 식별하는 작업은 시간이 오래 걸리고 노동력이 많이 필요한 과정이다. 본 논문에서 제안한 방법은 이러한 단점을 완화시킬 수 있으며, 사용자의 움직임을 기반으로 네비게이션 메쉬를 자동으로 생성하기 때문에 모든 과정을 손쉽게 업데이트 할 수 있다.



(a) (b)

Fig. 10. Movement of NPC with various navigation mesh

V. Conclusions

본 논문에서는 사용자의 경로를 분석하여 NPC의 이동경로를 제어할 수 있는 프레임워크를 제안했다. 특히, 같은 맵에서도 고정된 움직임이 아닌 사용자의 이동궤적에 따라 NPC의 움직임이 다양한 형태로 계산된다. 이 과정에서 사용자가 주로 다니는 골목을 끊어지지 않게 계산하기 위해 경로-유사도 기법을 제안하였으며, 이 기법을 통해 안정적으로 네비게이션 메쉬를 생성할 수 있었다. 모든 결과에서 활용한 메타데이터는 1,000개의 PC 경로를 활용했으며, 네비게이션 메쉬로 생성된 후로는 실시간으로 NPC들이 이동되는 결과를 보여주었다.

향후, 이 결과를 가상현실에 적용하여 사실적으로 사용자와 상호작용 할 수 있는 애플리케이션에 적용해볼 것이다. 뿐 만 아니라 실제 사용자가 길목을 탐색하는 과정은 위치 이외에 소리, 속도, 시점 등 다양한 물리적 속성이 반영되어야 하며, 향후 이러한 속성들을 고려하여 좀 더 정확하게 NPC의 움직임을 제어할 수 있는 방법으로 확장할 예정이다.

REFERENCES

[1] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, “Probabilistic roadmaps for path planning in high-dimensional configuration spaces,” IEEE transactions on Robotics and Automation, vol. 12, no. 4, pp. 566-580, 1996. DOI:10.1109/70.508439

[2] M. G. Choi, J. Lee, and S. Y. Shin, “Planning biped locomotion using motion capture data and probabilistic roadmaps,”ACM

Transactions on Graphics, vol. 22, no. 2, pp. 182-203, 2003. DOI:10.1145/636886.636889

[3] A. Kamphuis and M. H. Overmars, "Finding paths for coherent groups using clearance," ACM SIGGRAPH/Eurographics Symposium on Computer Animation, 2004, pp. 19-28. DOI:10.1145/1028523.1028526

[4] Y. Li and K. Gupta, "Motion planning of multiple agents in virtual environments on parallel architectures," in Proceedings 2007 IEEE International Conference on Robotics and Automation, 2007, pp. 1009-1014. DOI:10.1109/ROBOT.2007.363117

[5] J. Van Den Berg, S. Patil, J. Sewall, D. Manocha, and M. Lin, "Interactive navigation of individual agents in crowded environments," Interactive 3D Graphics and Games, 2008. DOI:10.1145/1342250.1342272

[6] J. Pettr´e, P. d. H. Ciechomski, J. Ma¨ım, B. Yersin, J.-P. Laumond, and D. Thalmann, "Real-time navigating crowds: scalable simulation and rendering," Computer Animation and Virtual Worlds, vol. 17, no. 3-4, pp. 445-455, 2006. DOI:10.1002/cav.147

[7] J. Pettr´e, H. Grillon, and D. Thalmann, "Crowds of moving objects: Navigation planning and simulation," In Proceedings IEEE International Conference on Robotics and Automation, 2007, pp. 3062-3067. DOI:10.1109/ROBOT.2007.363937

[8] A. Sud, E. Andersen, S. Curtis, M. Lin, and D. Manocha, "Real-time path planning for virtual agents in dynamic environments," IEEE Virtual Reality Conference, 2007, pp. 91-98. DOI:10.1109/VR.2007.352468

[9] Sud, A., Gayle, R., Andersen, E., Guy, S., Lin, M., Manocha, D.,"Real-time navigation of independent agents using adaptive roadmaps," ACM Symposium on Virtual Reality Software and Technology, vol. 1, pp. 99-106, 2007. DOI:10.1145/1315184.1315201

[10] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," IEEE transactions on Systems Science and Cybernetics, vol. 4, pp. 100-107, 1968. DOI:10.1109/TSSC.1968.300136

[11] A. Stentz, "Optimal and efficient path planning for unknown and dynamic environments," Carnegie-Mellon University Pittsburgh Pa Robotics Lab., Tech. Rep., 1993. DOI:10.1.1.15.3683

[12] B. Stout, "The basics of a* for path planning, game programming gems," 2000.

[13] C. Thompson, "Halo 3: How microsoft labs invented a new science of play," Wired Magazine, vol. 15, no. 9, pp. 15-09, 2007.

[14] E. Guzman, "Valve thrills fps fans with new counter-strike game," 2011.

[15] I. T. A. Havok, "en lınea," Intel Corporation, vol. 15.

[16] G. Comeau and A. Paramonoff, "Data path engine," Dec. 6 2001, US Patent App. 09/871,481.

[17] Marek Kopel, Tomasz Hajas, "Implementing AI for Non-player Characters in 3D Video Games," Intelligent Information and Database Systems, pp. 610-619, 2018. DOI:10.1007/978-3-319-75417-8_57

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. Prof. Kim is an assistant professor in the department of software application in Kangnam University. His current research interests include fluid animation and virtual reality.