

동적 환경에서 그룹 이동을 위한 경로 계획

유건아*, 조수진**

Path-Planning for Group Movement in Dynamic Environments

Kyeonah Yu *, Su-Jin Cho **

요약

가상의 캐릭터가 이동하기 위해 경로를 계획하는 일은 컴퓨터 게임을 포함한 여러 응용 분야에서 필수적인 문제이다. 기존의 로봇 경로 계획과는 달리 하나 이상의 캐릭터가 그룹 이동을 하는 경우가 많으며 이 경우에는 이동 경로의 최단 여부보다는 자연스러운 그룹 이동 등의 질적인 면이 강조된다. 본 논문에서는 리더로 정해진 단일 캐릭터에 대해 정적인 환경에서의 전역 경로를 계획하고 이 경로를 따라 그룹이 이동하며 동적 장애물을 피하도록 지역 경로를 계획하는 2단계 경로 계획 방법을 제안한다. 그룹이 이동할 수 있는 공간은 리더를 중심으로한 정방형 격자 배열을 이용하여 확보하고 이를 격자창이라고 부른다. 멤버 캐릭터들은 격자창 공간 안에서 리더에 대한 상대적인 위치를 잡아 이동하며 격자창이 이동할 수 있는 경로를 계획하기 위해 정적 환경을 격자창에 대한 형태 공간으로 치환하여 로드맵을 구한다. 지역 경로 계획 단계에서는 리더에 대해 인공 포텐셜 필드를 이용하여 동적 장애물을 회피하도록 하고 나머지 멤버들에 대해서는 격자창 내에서 리더에 대한 상대적인 위치를 확보하는 방법으로 연산량을 줄인다. 제안하는 경로 계획 방법을 구현하기 위한 효율적인 알고리즘을 소개하며 정적 환경에서 계획된 경로를 따라 그룹이 이동하면서 동적 장애물에 대해 효과적으로 대처하는 것을 시뮬레이션을 통해 보여 준다.

▶ Keywords : 경로 계획, 그룹 이동, 동적 장애물, 포텐셜 필드

Abstract

Path planning is an essential problem to make virtual characters navigate in many applications including computer games. In many cases, multiple characters move in a group and qualitative aspects of planned paths are emphasized rather than optimality unlike Robotics. In this paper, we

• 제1저자 : 유건아 • 교신저자 : 유건아

• 투고일 : 2012. 11. 29, 심사일 : 2012. 12. 31, 게재확정일 : 2013. 1. 13.

* 덕성여자대학교 컴퓨터학과(Dept. of Computer Science, Duksung Women's University)

** 덕성여자대학교 전산정보통신대학원(School of and Computer and Data Communication, Duksung Women's University)

※ 본 연구는 2011년도 덕성여자대학교 교내연구비 지원에 의해 수행되었다.

propose a two-level path planning algorithm in which the global path is planned for a single character specified as a leader and then the local path is planned to avoid dynamic obstacles while the group following this path. The space for group movement is achieved in the form of square grid array called a grid window. Member characters are located relatively to the leader within a space and moved. The static environment is reduced to the configuration space of this grid window to generate a roadmap on which a grid window can move. In local path planning, only the leader avoids dynamic obstacles by using an artificial potential field and the rest of members are located relatively to the leader in the grid window, which reduces computational load. Efficient algorithms to implement the proposed planning methods are introduced. The simulation results show that a group can handle with dynamic obstacles effectively while moving along the planned path for a static environment.

▶ Keywords : Path Planning, Group Movement, Dynamic Obstacles, Potential Field

I. 서 론

경로 계획은 캐릭터가 주어진 시작 위치로부터 목표 위치까지 이동하는 길을 찾는 문제로 과거 수십년 동안 로봇틱스 분야에서 활발하게 연구되어 온 문제이다. 가상 환경에서의 경로계획에서는 기존의 로봇 경로 계획과는 달리 자연스러움이 강조되는데 특히 다수의 캐릭터가 이동할 때에는 경로 자체의 자연스러움도 중요하지만 이동하는 캐릭터간의 상대적 위치의 유연함이나 동적 장애물에 대한 멤버 각각의 지역적 수정 방식도 자연스러움의 중요한 결정 요소이다[1][2]. 본 논문에서는 이와 같은 요구 사항을 만족할 수 있는 경로 계획 방법을 제안한다. 제안하는 방식의 특징은 리더에 대해서는 정적 환경에서의 전역 경로 계획과 동적 장애물에 대한 지역 경로 계획의 두 단계로 경로를 계획하고 그룹 멤버들에 대해서는 상태가 없이 구현된 보이드(boi)[3]와 유사하게 리더에 상대적으로 행동하게 하여 신속하면서도 계획된 이동을 하는 것처럼 보이게 하는 것이다. 이를 위해 그룹 전체를 포함하는 격자창(grid window)을 정의하고 이 격자창이 이동할 수 있도록 전역 경로를 계획하도록 한다.

전역 경로 계획에서는 격자창이 지나갈 수 있는 공간을 형태 공간(configuration space) 표현을 이용하여 미리 확보하고, 그 위에 가시성 그래프(visibility graph)를 생성하여 로드맵으로 이용하도록 한다. 가시성 그래프는 게임 분야에서 캐릭터의 이동을 위해 많이 사용되는 웨이포인트를 자동으로 생성할 수 있고 최단 경로를 생성할 수 있다는 장점이 있어

게임 분야의 경로 계획 방식으로 주목받는 로드맵 방식이다[4][5]. 그러나 노드가 장애물의 꼭지점에 생성되어 계획된 경로가 장애물과 밀착된 형태가 되어 그룹 이동에는 적합하지 않다. 본 논문에서는 작업 공간(workspace)을 격자창에 대한 형태 공간을 치환하여 그 위에 가시성 그래프를 생성함으로써 이를 해결하고자 한다.

이동 단계에서는 계획 단계에서 생성된 기간 경로(backbone path)를 따라 이동하면서 동적으로 발생하는 장애물을 피하는 지역 경로 계획을 한다. 지역 경로 계획에는 동적 장애물 환경에서 많이 사용되는 로봇 경로계획 방법인 인공 포텐셜 필드를 이용한다. 인공 포텐셜 필드는 정적 및 동적 환경, 단일 혹은 그룹에 대해 구분없이 활용될 수 있는 장점이 있는 반면 전역적인 최적 경로를 찾지 못하고 연산량이 많다는 단점이 있다[6][7]. 그러므로 본 논문에서는 인공 포텐셜 필드를 지역적으로, 리더에 대해서만 이용하고 나머지 멤버들에 대해서는 격자창 내에서 리더에 대한 상대적인 위치를 확보하는 방법으로 연산의 효율성을 꾀한다.

제안한 경로 계획 방식을 효율적으로 구현하기 위해 격자창 정의, 형태 공간 생성, 가시성 그래프 생성을 연산 기하학 분야의 알고리즘을 응용하며 시뮬레이션을 통하여 경로 계획의 결과와 이동 모습을 보여 준다. 또한 계획된 전역 경로를 따라 이동할 때 동적 장애물이 나타나면 리더에 대해 인공 포텐셜 필드를 적용하여 우회하고 나머지 멤버들은 정해진 창안에서 리더에 대해 상대적으로 움직이며 동적 장애물을 피하는 것을 보여 준다. 이와 같이 함으로써 전체 멤버에 대해 포텐셜 필드를 적용하는 기존의 방식보다 연산량을 줄일 수 있음을 정량적으로 확인한다.

본 논문의 구성은 다음과 같다. 2장에서는 기존의 게임 분야에서 경로계획에 관한 연구를 조사하고 3장에서는 기간 경로를 구하는 전역적 경로 계획 방법과 이동하면서 진행하는 지역적 경로 계획 방법을 살펴본다. 4장에서는 구현 방법과 시뮬레이션 결과에 대해 살펴보고 5장에서 결론을 맺는다.

II. 관련 연구

게임 개발 분야에서 가장 일반적인 경로 계획 방법은 배경을 그리드 기반으로 나누고 빈 그리드에 대해 A* 탐색 방법을 적용하는 것이다. 이 방법은 경로가 존재하면 반드시 찾고, 찾은 경로는 최적이라는 장점이 있지만 세밀한 표현을 위해 그리드의 수가 많아지면 탐색 시간이 오래 걸리고 경로가 지그재그 형태가 되므로 자연스러운 경로를 위해 평활화 과정이 요구되는 단점이 있다[8][9]. 다른 방법으로는 자유 공간에 웨이포인트를 두어 역시 A* 탐색 방법에 의해 이들을 연결하는 최단 경로를 찾는 것이 있는데 이는 레벨 설계자의 의중에 따라 웨이포인트의 위치가 결정되고 동적 변화에 대해 대처할 수 없다는 단점이 있다[10]. 이에 대한 대안으로 자동으로 웨이포인트를 설정하는 방법이 있는데 가시성 그래프(visibility graph)가 이 중 하나로 이용되고 있다. 가시성 그래프는 로봇틱스 분야에서 로봇 경로 계획을 위해 사용되던 방법으로 장애물의 꼭지점들을 웨이포인트로 하기 때문에 기존의 그리드 방법에 비해 탐색 공간이 작고 웨이포인트를 자동으로 설정할 수 있을 뿐 아니라 웨이포인트들을 연결한 링크들을 이용하여 최단 경로를 보장한다는 장점이 있다[4][11]. 그러나 장애물의 꼭지점을 이용하기 때문에 이동하는 경로의 여유공간은 전혀 없으므로 그룹 이동에 바로 적용하는데 문제가 있다. 본 논문에서는 가시성 그래프의 장점을 취하면서 그룹 이동에 적용할 수 있도록 실제 장애물에서 격자창의 부피만큼 오프셋된 가상 경계에 웨이포인트를 두는 방법을 사용한다. 이와 같이 생성된 가시성 그래프 위의 경로는 그룹이 이동할 수 있는 공간을 확보할 수 있게 하여 캐릭터들은 격자 창 내에서의 랜덤한 움직임으로 자연스러운 대형을 유지하면서 이동할 수 있도록 한다.

가시성 그래프를 응용하여 자연스러운 경로 생성을 시도한 시도는[12]에서도 찾아볼 수 있다. [12]에서는 경로의 자연스러움을 정의하고 장애물로부터의 여유 공간을 확보하기 위해 장애물을 확장하고 이에 따라 사라지는 자유공간을 보상하기 위해 보로노이 링크를 이용한 vv-복합체(visibility-voironi complex) 구조체를 제안하였다. 이 위에서 A* 혹은 다익스트라 알고리즘을 이용하여 구해진 경로

를 구하는데 그룹 이동과 동적장애물에 대해 다루지 않는다. vv-복합체는 복도 지도(corridor map) 방식[1][13]으로 확장되어 단일 혹은 그룹에 대해 요구되는 최소 여유 공간을 가진 경로를 계획할 수 있도록 하고 그 안에서 동적 장애물을 피하면서 이동할 수 있도록 하였다. 이 방식에서의 주안점은 자연스러운 경로의 생성이다. 복도 지도 방식은 동적 장애물을 만났을 때에도 피하는 경로도 자연스러움을 유지하기 위해 서브복도(subcorridor)를 생성하는데 복도 지도는 경로 계획의 주요핵심이며 구성이 복잡하여 생성과정이 많은 비중을 차지한다. 본 연구에서는 전역 경로의 자연스러움은 보장하지만 동적 장애물을 회피하기 위한 지역 경로는 전역 경로의 자연스러움을 강조할 필요가 없다는 점에 착안하고 복도 지도보다 간단한 로드맵 방식을 제안하는 것이다. 즉, 기존의 가시성 그래프 링크에 가상의 격자창이 더해지는 방법으로 로드맵의 생성이 구성이 간단하면서도 그룹과 동적 장애물에 대한 처리가 가능하도록 한다.

동적 환경에서 경로 계획을 위해 인공 포텐셜 필드가 널리 사용되었다. 그러나 [6]을 포함한 많은 문헌에서 지적되었듯이 인공 포텐셜 필드의 단점은 지역 최적(local optima)에 빠질 수 있다는 것이다. 노이즈 삽입의 랜덤방식 혹은 이전 위치를 기억하는 학습 방식과 같이 직접적으로 지역 최적 상태를 벗어나도록 해결하는 접근 방식도 있고 인공 포텐셜 필드를 전체 경로 방식이 아닌 지역적 수정 방식으로 택하여 해결하는 접근 방식도 있다[14][15][16]. 이들이 택한 전역 경로 계획을 계획하는 방식은 모두 본 연구와 다를 뿐 아니라 본 연구에서는 리더 하나에 대해서만 지역적 수정을 하기 때문에 그룹을 처리하는 방식에 있어서도 차별화된다.

게임 분야에서 그룹 이동에 가장 많이 이용되는 방법은 [3]의 플로킹으로 불리는 조타행동일 것이다. 플로킹은 개개 보이드의 행동을 간단한 지역적 규칙을 이용하여 제어하는 방식으로 여러 가지 형태의 조타 행동이 정의되어 있다. 이들을 단독으로 혹은 조합하여 사용하면 오픈 환경에서 개개의 보이드들은 매우 자연스럽게 움직인다. 하지만 계획이라는 개념이 없이 지역적으로 움직이기 때문에 전체적 경로 생성이 불가능하고 환경에 따라 쉽게 벗어날 수 없는 상태에 빠지기도 한다.

그러므로 전역적으로 최적에 가까운 경로를 계획하여 지역적 최적에 빠지지 않고도 동적 장애물에 실시간으로 대처하며 자연스러운 경로 이동을 할 수 있도록 가시성 그래프에 기반한 로드맵 방식과 인공 포텐셜 필드를 접목한 2단계 경로 계획 방법을 제안한다.

III. 2단계 경로 계획

본 연구의 접근 방식을 흐름도로 나타내면 그림 1과 같다. 계획 단계에서는 정적 장애물만 대상으로 전역 경로를 생성하게 되고 생성된 전역 경로에 따라 이동하는 단계에서 계획 단계에는 존재하지 않던 동적 장애물을 만나 지역적으로 경로를 수정하게 되는 지역 경로 계획을 수행하게 된다.

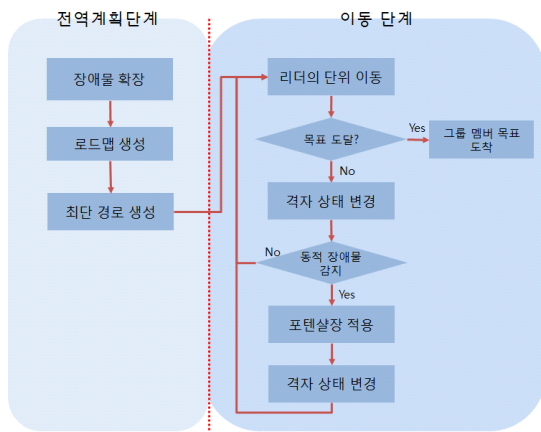


그림 1. 경로 계획 흐름도
Fig. 1. Path Planning Flowchart

1. 전역 계획 단계(Global Planning Phase)

기간 경로(backbone path) 계획의 중심 아이디어는 그룹 이동을 위한 격자창의 사용이다. 그룹 멤버들이 모두 포함되는 격자창을 정의한 후에는 격자창이 움직일 수 있는 자유 공간을 형태 공간(configuration space) 방식을 이용하여 표현하고 그 위에 경로를 계획한다.

1.1 형태 공간 장애물

격자창의 크기와 모양은 문제에서 요구하는 사양에 따라 유연하게 정해질 수 있다. 그림 2에서 아홉 멤버로 이루어진 그룹에 대한 격자창의 예를 보여준다. 그룹 멤버간의 필요한 여유 공간에 따라 격자창의 크기를 결정할 수 있으며 자유공간에서 주어진 통로의 너비에 따라 직사각형 모양의 격자창이 사용될 수도 있다. 그러나 자연스러운 경로를 위한 로드맵 생성을 위해서 사용하는 장애물 확장의 오프셋가 격자창 외접원의 반지름이고 이 반지름이 최소가 되는 경우가 정사각형 모양이므로 정사각형 격자창이 선호된다.

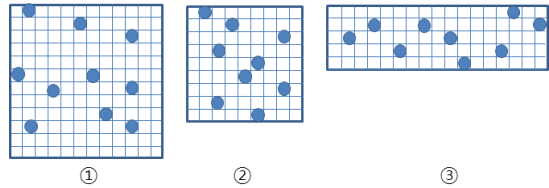


그림 2. 아홉 멤버를 포함한 격자창의 종류
Fig. 2. Types of Grid Window including nine members

장애물이 존재하는 작업 공간(work space)에서 격자창이 이동하도록 계획하기 위해서 장애물과의 충돌 여부를 계산하는 것이 시간 소모가 큰 작업이므로 격자창을 점으로 간주하고 장애물들을 그에 대응하여 확장하는 개념인 형태 공간으로 문제를 매핑한다. 격자창을 G 라고 하고 전체 형태 공간을 C 라고 하면 작업 공간의 장애물 O_i 에 대한 형태 공간 장애물 CO_i 는 다음 식 (1)과 같이 정의되며 n 개의 장애물이 존재하면 전체 형태 공간에서의 자유공간 C_{free} 는 식 (2)와 같이 정의된다.

$$CO_i = \{q \in C \mid G(q) \cap O_i \neq \emptyset\} \quad (1)$$

$$C_{free} = C - \bigcup_{i=1}^n CO_i \quad (2)$$

그림 3에서 내부에 채워진 도형들이 작업 공간의 장애물이며 도형 윤곽선이 격자창의 크기에 따라 확장된 형태 공간 장애물이다. 격자창의 크기에 따라 형태 공간 장애물의 생성에 차이가 있어 격자창 ①에 대해 생성된 형태 공간 장애물의 경우 자유공간의 일부가 막히고(그림 3의 좌측) 격자창 ②에 대해서는 그렇지 않은 것을 확인할 수 있다(그림 3의 우측).

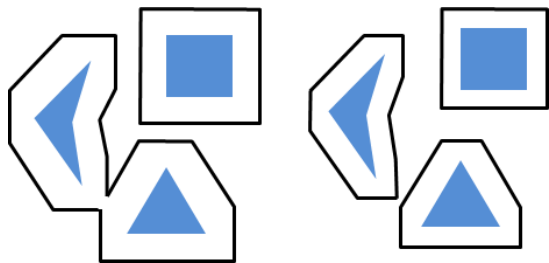


그림 3. 그림 2의 격자창 ①과 ②에 대한 형태 공간 장애물
Fig. 3. Configuration Space Obstacle for Grid Windows ① and ② in Fig. 2

1.2 가시성 그래프와 경로 탐색

자유 공간(C_{free})에서 경로를 찾기 위해 최단 경로 탐색이 보장되어 있는 가시성 그래프(visibility graph)를 로드맵으로 이용한다. 가시성 그래프는 이동의 시작 위치와 목표 위치, 장애물의 꼭지점들을 그래프의 노드로 하고 노드와 노드를 연결하는 선분 중에 장애물과 교차하지 않는 선분을 그래프의 링크로 하여 형성된다. 최단 경로를 찾기 위한 로드맵으로 사용되는 가시성 그래프는 일반적으로 축소형(reduced) 가시성 그래프이다. 축소형 가시성 그래프란 가시성 그래프의 링크 중에서 오목 꼭지점을 끝점(endpoint)으로 하는 링크를 제거한 그래프를 말하는데 오목 꼭지점을 포함하는 링크는 절대 최단 경로를 구성하지 않기 때문에 최단 경로를 찾기 위해서는 축소형 가시성 그래프를 이용하는 것이 타당하다[17]. 축소형 가시성 그래프를 이용하면 그래프를 생성하는데 소요되는 시간과 경로를 탐색하는 시간 모두를 절약할 수 있을 뿐 아니라 링크의 연결이 둔각으로 이루어져 경로에서 급격한 방향 전환(sharp turn)을 제거할 수 있다. 축소형 가시성 그래프가 얻어지고 나면 전역 경로 계획의 마지막 단계로 A* 알고리즘을 이용하여 최단 경로를 찾는다. 이 때 휴리스틱으로는 현재 노드에서 목표 노드까지의 직선거리를 나타내는 유클리디언 거리를 사용한다.

그림 4는 오목 꼭지점으로부터 구성된 링크를 제거하고 형태 공간 장애물 위에 형성된 가시성 그래프의 예를 보여 준다.

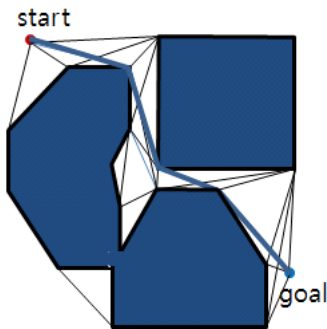


그림 4. 그림 3의 형태 공간 장애물에 대한 가시성 그래프와 최단 경로
Fig. 4. Visibility Graph and the Shortest Path for Configuration Space Obstacles in Fig.4

2. 이동 단계(Moving Phase)

이동 단계에서 리더는 방향과 속도를 갖고 A* 알고리즘을 통해 얻은 경로의 중간 목표를 경유하여 최종 목표에 다다를

때까지 이동하며 이동할 때마다 동적 장애물의 존재 여부를 검사하여 지역 경로 계획을 수행한다. 격자창내의 다른 멤버들은 리더에 상대적으로 움직여 자연스럽게 무리를 유지하며 이동한다.

2.1 격자창내의 이동

그룹 멤버들의 이동은 격자창 내에서 리더에 대한 상대적인 위치를 변화시켜 얻어지기 때문에 일단 리더의 이동이 결정되면 나머지 멤버들의 이동은 기간 경로(backbone path)를 따라 움직이는 경우나 포텐셜 필드에 의해 움직이는 경우에 관계없이 일정하다. 리더는 전체 격자를 관리하며, 아래와 같은 검사를 실시하여 이동이 가능한 셀과 불가능한 셀을 판별하게 된다. 검사의 우선순위는 다음과 같다.

- | |
|---------------|
| 1. 다른 멤버의 존재 |
| 2. 정적 장애물의 존재 |
| 3. 동적 장애물의 존재 |

격자창내의 각 셀은 상태값을 가지고 있어 위의 검사 결과에 따라 장애물이나 개체가 셀에 존재하지 않은 경우에 상태값을 0으로 두고 장애물이나 이웃이 이미 셀에 존재하고 있다면 그 상태값은 0이 아닌 다른 값으로 설정한다. 각 멤버는 격자 내에서 그림 5와 같이 현재 셀과 인접한 상하좌우, 그리고 대각선의 셀을 검사하여 상태값이 0인 셀을 선택하여 다음 위치로 지정하게 된다.

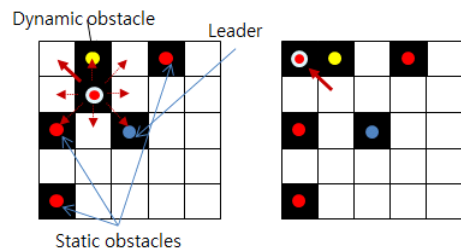


그림 5. 격자창의 상태와 멤버 이동
Fig. 5. State of Grid Window and Member Movement

위와 같은 과정을 통해 얻은 이동 가능한 셀만을 검사하여 멤버의 다음 위치를 알아내면 되기 때문에, 매번 모든 멤버들이 동적 장애물과 이웃 멤버를 검사하지 않게 되어 계산의 양을 줄일 수 있으며 격자창의 상태를 가지고 있지 않아 최소의 상태만을 저장하고도 리더를 따라 지능적으로 움직이는 것처럼 보이게 하는 장점이 있다.

2.2 동적 장애물 충돌 회피

리더는 격자의 기준점이 되는 멤버로, 방향과 속도를 가진다. A* 알고리즘을 통해 얻은 경로의 중간 목표(가시성 그래프의 노드)를 경유하여 이동하며, 최종 목표에 다다르게 되면 탐색을 중지하게 된다. 이동할 때마다 동적 장애물과의 충돌 여부를 검사한다. 만일 동적 장애물을 만나지 않는다면 리더의 상태는 0으로, A* 알고리즘을 통해 얻은 경로로 이동하며, 동적 장애물과의 충돌을 감지한다면 상태값을 1로 변경하여, 포텐셜 필드를 이용하여 동적 장애물을 회피하여 이동하게 된다. 동적 장애물과 일정 거리를 두게 된다면, 다시 상태값은 0으로 변경되어 원래의 경로로 이동하게 된다.

지역 경로계획에서는 목표로부터 발생하는 끌힘(attractive force)과 장애물로부터는 발생하는 밀힘(repulsive force)의, 이 두 값을 이용해 개체를 목표로 이르게 하는 포텐셜 필드를 이용한다. 목표로부터 발생하는 끌힘 F_a 와 장애물로부터 발생하는 밀힘 F_r 에 대한 식은 각각 다음과 같다.

$$F_a = F_{cg} \left[\frac{x_g - x_0}{d(g)} \quad \frac{y_g - y_0}{d(g)} \right]$$

$$F_r = \sum_i \frac{F_{cr}}{d^2(i)} \left[\frac{x_i - x_0}{d(i)} \quad \frac{y_i - y_0}{d(i)} \right]$$

F_{cg} 과 F_{cr} 는 각각 끌힘과 밀힘 상수로 실험을 통해 적절한 값을 지정해주게 되며 (x_g, y_g) 는 목표의 좌표, (x_0, y_0) 는 리더의 좌표, (x_i, y_i) 는 i 번째 장애물의 좌표를 나타낸다. $d(g)$ 는 리더와 목표 사이의 거리를, $d(i)$ 는 리더와 i 번째 장애물 사이의 거리를 나타낸다. 결과 힘 R 은 끌힘과 밀힘의 합($R = F_a + F_r$)으로 나타내지며 리더는 R 의 방향으로 이동 방향을 잡는다.

IV. 구현 및 시뮬레이션 결과

제한한 경로 계획 방식은 연산 기하학 알고리즘을 이용하여 효율적으로 구현하였으며 시뮬레이션을 통해 결과를 확인하였다.

1.1 전역 경로 계획 구현

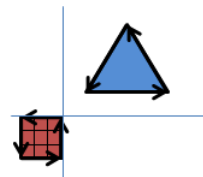
① 형태 공간 장애물 구현

격자창은 이동시에 회전하지 않으므로 형태 공간은 (x, y) 의 2차원 공간이 되며 형태 공간 장애물은 격자창을 W , i 번째 장애물을 O_i 라고 하면 형태 공간(C-공간) 장애물은 격자창의 민코스키 차와 장애물의 민코스키 합($CO_i = \ominus W \oplus O_i$)으로 정의된다. $CO = \bigcup_i CO_i$ 는[18]에서 소개하고 스타 알고리즘이라고 불리는 벡터 방식으로 구현하였다[19]. 그림 6은 스타 알고리즘에 의해 CO_i 를 구하는 과정을 보여준다. 장애물과 격자창의 모서리들을 벡터로 표현하고 각도에 따라 시계반대방향(CCW)로 정렬한 후 연결한 것인데 이와 같이 구해진 형태 공간 장애물의 합집합은 플레인 스위프(Plane Sweep) 알고리즘을 이용하여 $O(n \log n)$ 에 구한다. 또한 이 과정에서 가장 시간 소요(time-consuming)되는 작업인 arctan2를 사용하는 대신 의사(pseudo) 각도를 구하는 theta 알고리즘으로 효율을 꾀하였다[20].

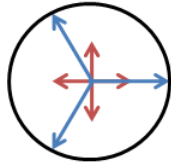
```
double theta(Node n1, Node n2) {
    double dx, dy, ax, ay, t, res;
    dx=n2.x-n1.x;   dy=n2.y-n1.y;
    ax=abs(dx);     ay=abs(dy);

    if(ax+ay==0)    t=4;
    else            t=dy/(ax+ay);
    if(dx<0)       t=2-t;
    else if(dy<0)  t=4+t;

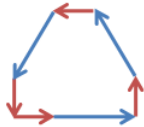
    res=t*90.0;    n2.setTheta(res);
    return res;
}
```



(a) 격자와 장애물의 모서리 벡터



(b) 모서리 벡터 시계반대방향으로 정렬



(c) 정렬된 벡터 연결하여 C-공간 장애물 생성

그림 6. 형태 공간 장애물 계산

Fig. 6. Computing Configuration Space Obstacle

② 가시성 그래프 구현

최단 경로를 찾기 위한 로드맵으로 사용되는 가시성 그래프는 일반적으로 축소형(reduced) 가시성 그래프이다. 축소형 가시성 그래프란 가시성 그래프에서 오목 꼭지점을 포함하는 링크를 모두 제거한 것으로 최단 경로 탐색에 영향을 주지 않고 링크수를 줄일 수 있다. 축소형 가시성 그래프는 회전 플레인 스위프 (Rotational Plane Sweep) 알고리즘을 이용하여 $O(n^2 \log n)$ 시간복잡도로 생성한다. 그림 7은 회전 플레인 스위프(Rotational Plane Sweep) 알고리즘을 이용하여 그래프를 생성하는 과정이다. 원으로 표시된 시작 노드에서 시계 반대 방향으로 회전하며 꼭지점과의 관계를 이진 트리로 표현하여 서로 보이는(visible) 여부를 $O(n \log n)$ 으로 검사하며 이를 $O(n)$ 개의 꼭지점에 대해 반복하여 $O(n^2 \log n)$ 이 걸린다.

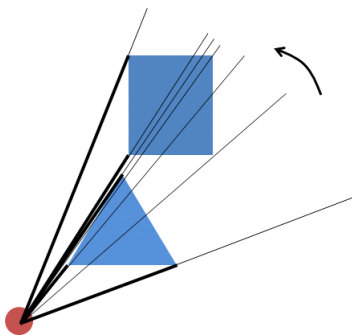


그림 7. 가시성 그래프 계산

Fig. 7. Computing Visibility Graph

③ 최단 경로 탐색 구현

생성된 가시성 그래프위에서 A* 알고리즘을 적용하여 최단 경로를 탐색한다. 휴리스틱으로는 유클리디언 거리를 이용하였고, 출발 노드에서 현재 노드까지의 실제 이동 비용인 g 와 현재 노드에서 목표 노드까지의 예상 비용인 h (휴리스틱)를 더한 값인 f 를 이용하여 가장 작은 비용이 드는 경로를 찾게 된다. 그림 8의 중간에 굵게 표시된 부분이 A* 알고리즘을 이용해 생성한 최단 경로이다.

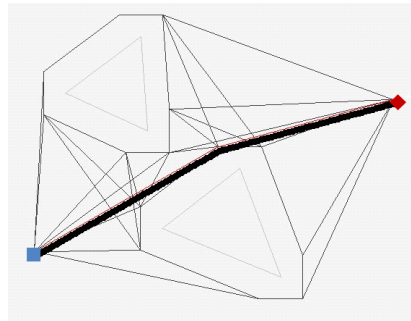


그림 8. 최단 경로 탐색

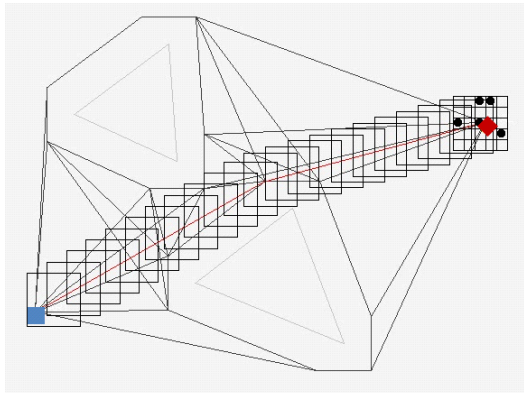
Fig. 8. Searching The Shortest Path

1.2 이동 시뮬레이션

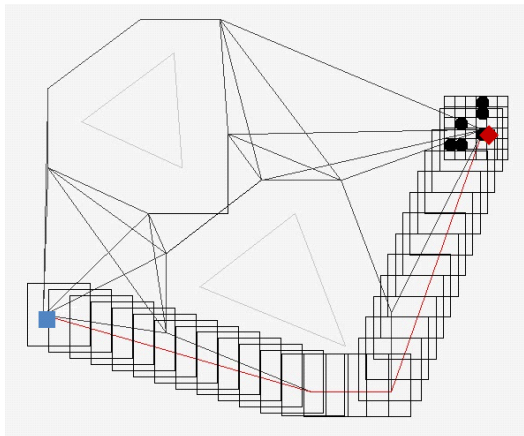
구현된 시뮬레이션 환경은 자바(Java) 언어로 구현하였고 Windows 7 환경에서 실험하였다. 맵 환경은 입의의 다각형으로 정적 장애물을 생성하여 제작할 수 있게 하였으며 앞 절에서 설명한 방식대로 형태 공간 장애물로 변환하여 최단 경로를 생성한 후, 격자창이 이동하는 것을 시뮬레이션하였다. 이동시에 나타나는 동적 장애물은 원으로 모델링하였다.

① 정적 환경에서의 경로 이동

그림 9는 정적 환경에서 찾은 경로로 A* 알고리즘을 이용해서만 이동하는 모습이며 격자창 크기에 따라 나타나는 결과를 보여주고 있는데 그림 9 (b)는 그림 9 (a)에 비해 큰 크기의 격자창을 가정한 경우이다. 큰 격자창에 대해 구한 C-공간 장애물이 작은 격자창에 대한 C-공간 장애물에 비해 크게 확대되므로 기존의 자유 공간이 일부 막혀 경로가 우회하는 모습을 보여 준다.



(a) 보이드 갯수 : 5 보이드 크기 : 10



(b) 보이드 갯수 : 6 보이드 크기 : 15

그림 9. 이동 시뮬레이션
Fig. 9. Moving Simulation

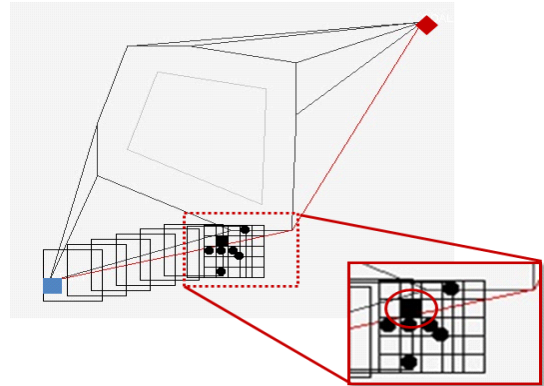
② 동적 환경에서의 경로 이동

그림 10은 동적 장애물을 만났을 경우에 대처하는 경로 이동 결과를 보여준다. 동적 장애물을 감지하지 않으면 A* 알고리즘에 의해 계획된 경로를 따라 이동하지만 동적 장애물을 만나면 리더에 대해 포텐셜 필드를 계산하여 국부적으로 경로를 수정하고 동적 장애물의 범위를 벗어나면 다시 이전에 찾은 최단 경로를 이동하는 모습을 볼 수 있다.

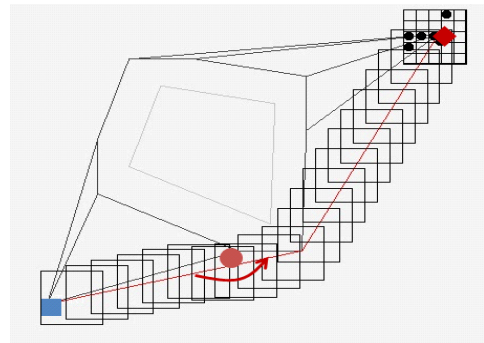
이 단계에서도 리더를 제외한 그룹 멤버는 격자창내에서 리더에 대한 상대적인 위치를 찾는 방식으로 이동을 하므로 리더에만 포텐셜 필드 알고리즘을 적용하여 그룹 멤버에 대한 포텐셜 필드 계산량을 줄일 수 있다.

그림 10의 (a)는 동적 장애물을 만났을 때, 동적 장애물이

포함된 격자를 검은색으로 표시하여 이동하지 못하는 공간을 나타내고 있다. (b)는 격자의 총 이동 경로로, 빨간색 화살표는 장애물을 피해가는 경로의 이동을 보여주고 있다.



(a)



(b)

그림 10. 동적 장애물 처리
Fig. 10. Dynamic Obstacle handling

③ 제안 방식의 성능 비교

본 논문에서 제안한 동적 장애물에 대한 처리 방식을 기존의 포텐셜 필드 방식과 성능 비교해 보았다. 장애물의 수와 위치를 변화시키며 얻은 서로 다른 5가지 환경에 대해 리더에만 적용하는 본 논문 제안 방식과 기존의 포텐셜 필드 방식으로 실행한 실행 시간을 표 1과 같이 msec단위로 측정하고 평균하였다. 이와 같은 실험을 이동하는 그룹 멤버의 수를 증가시키면서 반복한 결과가 그림 11과 같다. 그룹이 커짐에 따라 본 제안 방식에 의한 실행시간 증가는 미미하여 기존의 포텐셜 필드 방식과의 격차를 더 많이 벌어진다는 사실을 확인하였으며 본 논문에서 제안한 방식이 더 효율적이라는 결과를 얻을 수 있었다.

표 1. 실행시간 비교
Table 1. Comparison of Running Time

	Leader에 적용	모든 멤버에 적용
Case 1	8299	14214
Case 2	8081	13912
Case 3	14087	20728
Case 4	6805	12627
Case 5	8190	13832
Average	9092.4	15062.6

이 비교는 정적 장애물에 대해서는 A* 알고리즘을 적용하고, 동적 장애물에 대해 포텐셜 필드를 적용하는 방식에 대한 비교였으므로 전체 계획을 포텐셜 필드로만 하는 경우는 소요 시간이 더욱 커질 것이 자명하므로 본 논문에서 제안한 방식으로 지역 최적 문제를 해결할 수 있을 뿐 아니라 시간적으로도 효율적임을 확인할 수 있었다.

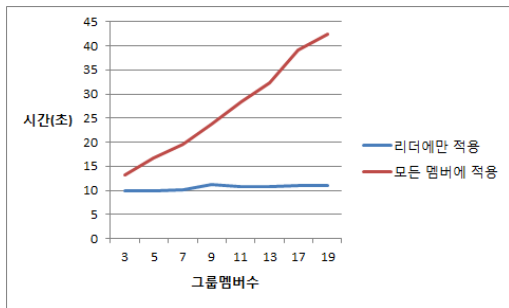


그림 11. 그룹 크기의 변화에 따른 실행시간 비교
Fig. 11. Comparison of Running Time as Group Size Changes

V. 결론

본 논문에서는 동적 장애물이 존재하는 환경에서 그룹이 이동하는 2단계 경로 계획 방법을 제안하였다. 2단계 경로 계획은 정적 환경에서 가시성 그래프 위에 A* 알고리즘을 이용하여 전역 경로를 계획하는 단계와 동적 장애물이 나타나면 포텐셜 필드를 이용한 지역 경로 계획을 통해 경로를 수정하는 단계로 구성되어 있다. 그룹이 이동하는 공간을 확보하기 위해 리더를 중심으로 한 격자창을 정의하여 멤버들의 자연스러운 이동을 가능하게 하였으며 이를 위한 상태 공간 표현과 최단 경로 생성 단계들은 연산 기하학의 효율적인 알고리즘들을 이용하여 효과적으로 구현되었다. 시뮬레이션을 통하여 생

성되는 경로와 경로를 따라 이동하는 과정을 확인하였으며 동적 장애물이 존재하는 경우의 지역적 수정도 확인하였다. 기존의 포텐셜 필드를 적용하던 방법과의 비교에서는 성능의 우수성도 확인할 수 있었다.

제안한 방식에서는 그룹을 포함할 수 있는 격자창이 움직일 수 있는 공간을 확보하기 위해 정적 장애물을 격자창에 대한 형태 공간 장애물로 변환하여 경로를 계획하는데 이 때 격자창이 큰 경우, 자유 공간이 막혀 경로가 크게 우회하는 경우가 발생하므로 이에 대한 보완책이 함께 연구되어야 한다. 그룹 이동의 다양성을 제공함으로써 이를 해결할 수 있는데 장애물 확장으로 인해 막힌 자유공간을 피해 가거나 격자창의 폭을 좁게 하여 지나가거나 하는 선택권을 부여하는 것이다. 이와 같이 상황에 맞는 이동 방식을 택하기 위해 추후에 캐릭터의 학습 기능에 대한 연구가 필요하다.

참고문헌

- [1] R. Geraerts, M. Overmars, "The corridor map method: A general framework for real-time high-quality path planning", *Computer Animation and Virtual Worlds* 18, pp. 107 - 119, 2007.
- [2] Y. Chen, Y. Lin, "Controlling the movement of crowds in computer graphics by using the mechanism of particle swarm optimization", *Applied Soft Computing* 9, pp1170 - 1176, 2009.
- [3] C. Reynolds, "Steering behaviors for autonomous characters", *Game Developers Conference*, pp 763-782, 1999.
- [4] X. Cui, H. Shi, "Direction Oriented Pathfinding in Video Games", *International Journal of Artificial Intelligence & Applications (IJAIA)*, Vol.2, No.4, October 2011.
- [5] H. Kim, K. Yu and J. Kim, "Reducing the Search Space for Pathfinding in Navigation Meshes by Using Visibility Tests", *Journal of Electrical Engineering & Technology*, Vol. 6, No. 6, pp. 867~873, 2011.
- [6] Y. Koren, J. Borenstein, "Potential field methods and their inherent limitations for mobile robot navigation", *IEEE International Conference on Robotics and Automation*, vol.2, Sacramento,

CA, pp. 1398-1404, 1991.

[7] J. Hagelbäck and Stefan J. Johansson, "A Multiagent Potential Field-Based Bot for Real-Time Strategy Games", International Journal of Computer Games Technology, Article ID 910819, 2009.

[8] Rabin, S. "A* speed optimizations and A* Aesthetic Optimizations," In: Deloura, M. (eds.): Game Programming Gems. Charles Rive Media, 264-287, 2000.

[9] Pinter, M. "Towards more realistic pathfinding," Game Developer Magazine April, 2001.

[10] Tozour, P. "Search Space Representations," In: Rabin, S. (eds.): AI Game Programming Wisdom 2. Charles Rive Media, 85-102, 2004.

[11] Young, T. "Expanded Geometry for Points-of-Visibility Pathfinding," In: Deloura, M. (eds.): Game Programming Gems 2, Charles Rive Media, 317-323, 2001.

[12] Wein, R., Berg, J., Halperin, D, "The Visibility-Voronoi complex and its applications", Annual Symposium on Computational Geometry, pp. 63 - 72, 2005.

[13] Geraerts, R., Overmars, M, "Enhancing corridor maps for real-time path planning in virtual environments", Computer Animation and Social Agents, 2008.

[14] R. Murphy, Introduction to AI Robotics, A Bradford Book: 1st edition, November 13, 2000.

[15] Hwang, Y.K., "A potential field approach to path planning", IEEE Transactions on Robotics and Automation, 8, Issue: 1 pp 23 - 32, 1992.

[16] J. Sheng, G. He, W. Guo, J. Li, "An improved artificial potential field algorithm for virtual human path planning", Proceedings of the Entertainment for education, and 5th international conference on E-learning and games, pp 592-601, 2010.

[17] J.C. Latombe, "Robot Motion Planning", KAP, pp. 310-317, 1991.

[18] T. Lozano-Perez, "Spatial planning: A

configuration space approach", IEEE Transactions on Computing, pp 108 - 120, 1983.

[19] S. M. LaValle, "Planning Algorithms", chapter 4, 2008.

[20] J. O'Rourke, Computational Geometry in C, Cambridge University Press, 1998.

저 자 소 개

유 건 아



1986: 서울대학교
제어계측공학과 공학사.
1988: 서울대학교
제어계측공학과 공학 석사.
1995: Univ. of Southern
California 컴퓨터학과 공학박사
현 재: 덕성여자대학교 컴퓨터학과 교수
관심분야: 인공지능,
경로계획 알고리즘
Email : kyeonah@duksung.ac.kr

조 수 진



2011: 덕성여자대학교
컴퓨터 공학과 공학사.
현 재: 덕성여자대학교
전산정보통신학과 석사과정.
관심분야: 게임 인공지능
Email : cho2460@duksung.ac.kr