복수의 동적 장애물에 대한 이동로봇의 최적경로설계

Optimal Path Planning of Mobile Robot for Multiple Moving Obstacles

김 대 광[1] · 강 동 중†

Dae Gwang Kim[1] · Dong-Joong Kang†

Abstract The most important thing for navigation of a mobile robot is to find the most suitable path and avoid the obstacles in the static and dynamic environment. This paper presents a method to search the optimal path in start space extended to time domain with considering a velocity and a direction of moving obstacles. A modified version of A* algorithm has been applied for path planning in this work and proposed a method of path search to avoid a collision with moving obstacle in space-tim domain with a velocity and an orientation of obstacles. The velocity and the direction for moving obstacle are assumed as linear form. The simulation result shows that a mobile robot navigates safely among moving obstacles of constant linear velocity. This work can be applied for not only a moving robot but also a legged humanoid robot and all fields where the path planning is required.

Keywords : Time-space state domain, Modified A* Algorithm, Moving and multiple obstacles, Path planning, Collision avoidance, Mobile robot.

1. 서 론

주행을 위한 이동로봇은 정적 및 동적 환경에서 장애물들을 회피하여 목표를 향한 최적의 경로를 발견하는 것이 최대 관심사 중의 하나이다. 특히 동적인 환경에서 로봇의 이동문제는 운동학 및 기구학을 고려해야하는 만큼 더욱 어려워진다. 로봇은 움직이는 사람 혹은 여러 물체들과 동적인 주변환경의 여러 상황을 지속적으로 인지하는 인식 시스템을 갖추고 있어야 충돌없이 목표를 향해 안전한 주행이 가능하다. 목표지향적인 이동로봇은 크게 두 가지 시스템을 갖추고 있다. 목표를 향해 주행하는 동안 국부 및 전역적 안전성을 유지하기 위해 주변의 상황을 끊임없이 관측하는 센싱 시스템과 이 시스템으로부터 획득된 지식을 이용하여 목표를 향해 안전하게 주행하기 위해 로봇의 경로를 계산하는 경로탐색 시스템으로 크게 나누어진다.

※ This work was supported by the IT R&D program of MIC/IITA [2006-S-028-01, Development of Cooperative Network-based Humanoids Technology].

† 교신저자 : 부산대학교 기계공학부 조교수
(E-mail : djkang@pusan.ac.kr)
[1] 부산대학교 기계공학부 석사과정(E-mail : dg_kim@pusan.ac.kr)

이러한 시스템들은 비단 일반적인 이동로봇(4 족 / 다지류 로봇, 바퀴를 사용하는 이동체)뿐만아니라 다른 대부분의 이동체(2 족보행로봇, 비행체, 수중체)에 대해서도 동일하게 적용될 수 있다. 최근에는 신뢰성 있는 휴머노이드 로봇의 개발이 많이 진척되었다. 이러한 로봇을 이용한 움직임 경로계획에 관한 연구는 국내외에서 최근 점차 활발해지고 있다. Kuffner와 Kanade 등은 휴머노이드 로봇의 발자국 경로계획에 관한 최근의 연구결과를 발표하였다[1]. 휴머노이드 로봇의 출발위치와 목표위치가 주어지고 장애물들로 채워진 환경에서 최적의 발자국 경로들을 구한다. 예상 가능한 움직임을 가진 이동장애물을 이용하여 발자국 경로계획을 수행하였다. 따라서 예상할 수 없는 장애물이 나타나는 동적인 환경에 대해서는 한계를 가진다. 경로계획은 잘 알려진 A* 알고리즘을 사용하였다[2]. 정해진 실행시간 내에서 주위환경에 대해 유효영역 유무에 대한 비용, 로봇의 다음 행동을 찾는데 드는 비용, 목표 도달까지 걸리는 비용들을 고려하여 발자국 경로를 탐색한다.

동적환경에서는 정적인 장애물 뿐만 아니라 다수의 작업로봇일 경우와 여러대의 움직이는 장애물에 대해 고려해야한다. 먼저 협동작업을 위한 멀티 로봇일 경우

작업시 서로 충돌이 발생하지 않도록 경로는 계획되어야한다. P. O'Donnell과 T. Lozano-Perez는 서로 근접한 두 대의 로봇에 대해 TC-다이어그램(Task-Completion Diagram)을 이용하여 로봇 상호간 충돌과 각각의 로봇이 서로 피하기를 기다리는 상황인 교착상태를 회피하는 방법을 연구했다[3]. TC-다이어그램은 여러 로봇이 자원을 공유하는 공간인 이미 알고있는 환경에서 탐욕적 알고리즘(Greedy algorithm)을 사용하여 충돌회피 스케줄링 문제를 분석한다. 각 로봇의 거리에 따른 개별 경로를 먼저 찾은 다음 각 경로에 대해 충돌이 발생할 때 TC－다이어그램을 그려 시간차를 두어 충돌을 회피한다. TC-다이어그램은 각 로봇들의 속도에 대한 고려는 제한했다.

동적환경에서 움직이는 장애물들이 복수일 경우 이동로봇과 이동하는 장애물들의 충돌을 회피하기 위해서는 다양한 파라미터들을 고려해야한다.

본 논문은 동적인 환경에 대한 상태공간에서 시간축으로의 탐색공간확장을 통해 속도와 방향을 가진 여러 대의 동적 장애물에 대한 최적의 이동경로를 구하는 방법을 제시한다. 경로계획은 A*알고리즘의 수정으로 구성한다. 논문의 나머지는 다음과 같다. 2 장에서는 복수의 장애물이 있는 환경에서 경로계획에 대한 기존 연구들을 기술하였고 3 장에서는 본 논문에서 사용한 경로계획 알고리즘에 대한 간략한 내용과 논문에서 제안한 시공간 확장에서 복수의 동적 장애물의 위치변화와 변형된 A*알고리즘 결합에 의한 속도를 가진 장애물과의 충돌을 회피할 수 있는 방법에 대해 기술한다. 4 장에서는 3 장에서 제안된 방법을 바탕으로 실험 결과 및 분석을 기술하였다. 마지막 5 장은 실험을 통해 얻을 수 있는 결론과 향후 과제에 대해 요약한다.

2. 기존 연구

동적 환경에서 이동로봇의 주행을 실세계에 적용하는 것은 상당히 어렵다. 로봇은 움직이는 사람, 여러 이동체들과 주변상황에서 임의적으로 발생하는 불확실성, 그리고 자신의 인식시스템의 한계를 극복하면서 목표지점까지 안전하게 주행할 수 있는 자신의 경로를 찾아야 한다.

경로계획에서 전역 접근들은 이동로봇의 현재 위치에서 목표지점까지 완전한 경로를 계산한다. 하지만 이 경우 주변 환경정보를 알고 있다고 가정한다. 이러한 방법들은 보통 예상치 못한 장애물들을 피하기 위해 국부적인 접근방법들을 사용하는 경우도 있다[4].

이동하는 장애물들을 가진 환경일 경우 한 가지 방법은 상태공간에 시간차원을 더하는 것이다[5]. 하지만 기존 연구에서는 장애물의 방향은 고려하지 않았다 . 또한 전역적인 접근 방법들은 목표지점까지 이동로봇의 경로에 대해 최적의 해결책들을 제공할 수 있지만 주된 약점은 완결되고 결정되어진 환경지식을 가정한다는 것이다. 실제 적용에서는 이러한 방법은 예측하지 못한 장애물들을 피하기 위해 국부적인 방법들을 결합한다[6,7].

반응적인 방법들로 불리는 이 방법들은 다음단계의 입력 제어만을 생성한다. 단지 가장 가까운 환경의 일부분만을 사용하여 현재의 센서 관측에 의해 실세계 모델을 갱신한다. DWA(Dynamic Window Approach)[6,8] 에서의 곡률 속도(curvature velocity)[9]와 선 곡률 방법(lane curvature velocity)[10]은 환경의 동적인 정보를 고려하지 않고 정적인 환경으로서 모든 장애물들을 간주한다.

다른 한편, 속도 장애물(Velocity Obstacles) 접근[11,12], 피할 수 없는 충돌 상태 개념(Inevitable Collision States concept)[13,14]등은 충돌 회피 신호를 계산하기 위해 장애물들의 속도에 관해 결정된 지식을 사용한다. 언급한 모든 방법들은 정적 및 동적 환경의 완전한 지식과 실세계의 결정된 표현에 의존한다.

베이시안 점유필터(Bayesian Occupancy Filter)[15]와 선형속도장애물(Linear Velocity Obstacles)[11]의 두 프레임웍을 결합한 방법은 인식과 로봇의 주행시스템 사이에 연결을 만들기 위해 확률적 구성을 기반으로 반응적 장애물을 회피한다[16].

확률적 속도 장애물 접근법(PVO)은 속도의 불확실성 추정과 원형장애물들의 반경에 대해 VO를 확장함으로써 제안되었다[16]. 여기서는 일반적인 센서 시스템으로부터 제공되는 동적 점유 그리드와 제한된 방법을 결합했다. 센서들은 로봇 주위의 점유된 공간과 자유공간의 확률적 추정과 물체들이 이동하면서 가지는 속도의 확률적 추정을 제공한다. 공간과 속도에 대한 관측을 통해 4D 확률적 점유그리드를 갱신한다[17]. 시간에서 충돌확률은 각 로봇의 도달된 속도에 대해 평가된다.

3. 동적 장애물 회피 알고리즘

동적인 환경에서 움직이는 이동체나 장애물에 대해 충돌을 피할 수 있는 최적의 이동경로 탐색은 아주 중요

한 문제이다. 안정적이고 최적인 경로를 탐색하기위해 본 연구에서는 A*알고리즘의 변형된 버전을 사용한다. 이 방법은 출발지점과 목표지점을 입력하면 탐색하고자 하는 환경에 대해 최적의 이동경로를 되돌려 준다. 본 논문에서 제안한 시간축으로 확장된 공간에서 속도를 가진 장애물의 움직임을 모델링하고 A*알고리즘에 의한 시공간 탐색에 의해 동적인 환경 하에서도 안정적이고 최적화된 로봇의 이동경로를 얻을 수 있다. 먼저 A*알고리즘의 간략한 내용에 대해 살펴본다.

3-1 A*알고리즘

A*알고리즘은 출발노드로부터 목표노드까지의 최적경로를 탐색하기 위한 것이다. 이를 위해서 각각의 노드에 대한 평가함수를 정의한다.

$$f(n) = g(n) + h(n) \tag{1}$$

여기서 $g(n)$ 은출발노드로보터 현재노드 n 까지의 경로비용을 나타내고, $h(n)$은 노드 n으로부터 목표노드까지의 경로비용을 나타낸다. 그러나 $h(n)$은 아직 탐색하지 않은 경로이므로 정확히 계산하기 어렵다. 따라서 경험적 규칙이 사용된다. 경험직 지식에 의한 $h(n)$에 대한 예측값을 $\hat{h}(n)$이라 하면, 노드 n에 대한 평가함수는 식(2)와 같다

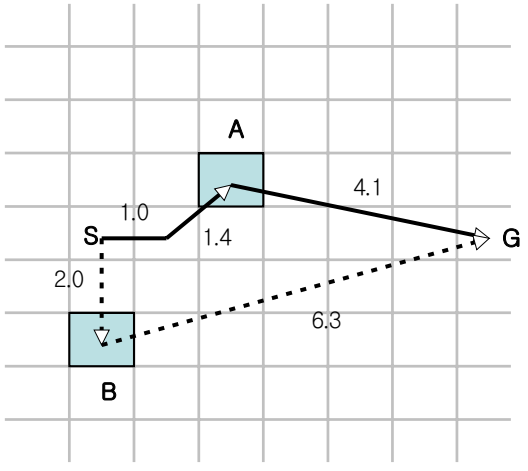


그림 1. A*알고리즘의 발견적 값 계산

$$\hat{f}(n) = g(n) + \hat{h}(n) \tag{2}$$

여기서 $g(n)$은 식(1)과 같고 $\hat{h}(n)$은 노드 n으로부터 목표노드까지의 예측 경로비용이다. 그림 1 은 출발점 S에서 목표점 G까지의 최적경로비용을 찾는 과정을 설명한다. A와 B는 A*알고리즘의 OPEN 리스트 내에 존재하는 임의의 두 상태노드(로봇의 위치)를 의미한다. 실선은 첫 번째 경로이다. 출발점에서 A위치까지의 경로 거리는 2.4이다. A에서 목표까지의 경로는 알 수 없으므로 직선거리로 간주한다. 직선의 길이는 4.1이다. 결론적으로 A의 경로비용은 6.5이다. 점선은 B의 경로이다. 출발점에서 B위치까지의 거리는 2.0이다 실선은 A까지의 거리보다 가깝다. 그러나 B에서 목표까지의 거리는 6.3이고, 전체 A*알고리즘의 발견적 값은 8.3이다. 따라서 A*알고리즘에서는 B보다는 A가 비용면에서 더 우월하다는 것을 보여주므로 다음 단계에서 후계노드로 확장될 후보노드는 B노드가 될 것이다. 표1에서 A*알고리즘을 나타내었다. 먼저 출발노드를 OPEN리스트에 넣고 알고리즘을 수행하여 목표지점까지 최적의 경로를 찾는 것이 알고리즘의 목적이다. OPEN리스트에서 하나의 노드를 꺼내어 후계노들을 생성한 후 각 후계노드들의 평가함수 값을 계산하여 각 리스트들에 넣는다. OPEN 리스트 목록이 빌 때까지 알고리즘은 반복된다. (OPEN은 후계노드를 파생하지 않은 노드 리스트, CLOSE는 후계노드를 파생한 노드 리스트이다. OPEN과 CLOSE는 큐(Queue) 자료구조형으로 구현된다).

표 1. A*알고리즘

A* algorithm($Goal_{pos}$, $Start_{pos}$)

```
// Initialize_Queu Start_pos e(Q_open, Q_dose);
// Initialize_Value();
  s_init ← Start_pos, Goal_pos ;
  Q_open.insert(s_init);
  while Q_open != 0 do
    s_cs ← Q_open.pop( );
    Q_dose. insert(s_cs) ;
    if GoalReach( s_cs ) then
       Return robot_path;
    end
    foreach s_suc ∈ s_all suc do
      s_suc ← Extract_Succ( s_cs) ;
      c_l ← LocationCost( s_suc ) ;
      c_h ← HeuristicCost(s_suc, Goal_pos) ;
      Q_open.insert(s_suc, s_suc. cost +c_l+ c_h) ;
    end
    Q_open. Sort();
  end
```

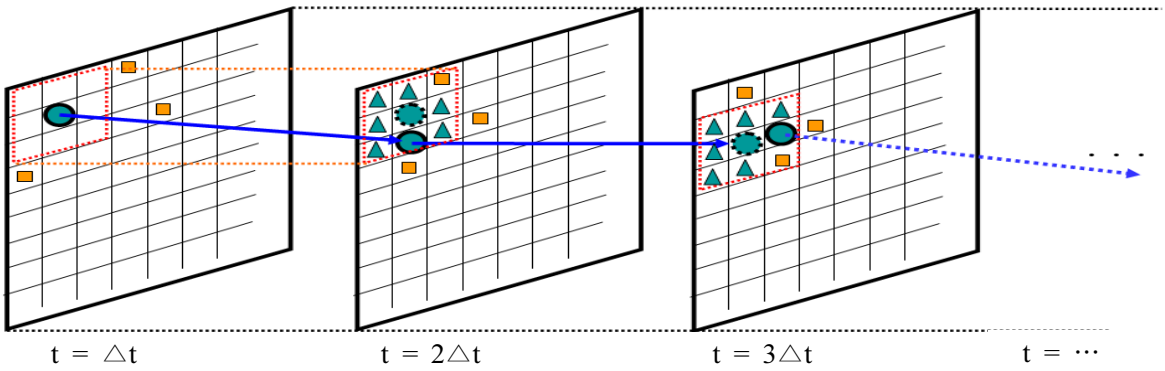


그림 3. 시공간 영역탐색(녹색원 : 이동로봇, 주황색사각형 : 이동장애물, 녹색삼각형 : 확장가능노드, 파란색실선 : 로봇의 이동경로, 주황색점선 : 로봇의 현재상태노드의 확장 범위, $t=\triangle t$에서의 로봇의 확장노드는 $t=2\triangle t$에서 선택함, 이동로봇은 확장되는 범위 중 한 곳으로 이동한다)

먼저, 출발위치 $Start_{pos}$와 도착위치 $Goal_{pos}$를 가지는 초기상태노드 s_{init}를 OPEN 리스트에 넣는다. 이 초기상태노드는 출발노드가 된다. OPEN 리스트에서 출발노드를 꺼내어 현재상태노드인 s_{cs}에 넣고 CLOSE 리스트에 s_{cs}를 추가한다. 그 노드가 목표인지 검사 후 맞으면 발견된 이동경로를 되돌려준다. s_{cs}가 목표가 아니라면 s_{cs}에 대한 모든 후계상태노드집합인 s_{allsuc}에서 주행가능한 모든 후계노드 s_{suc}에 대해 비용을 계산한다. 출발부터 현재노드까지 거리비용 c_l 과 s_{suc}로부터 목표지점까지의 휴리스틱비용 c_h를 더해서 식(2)의 값을 계산하고 그 결과를 중복을 피해 OPEN 리스트에 넣는다. 노드의 확장이 완료되면 OPEN 리스트에 대해 가장 작은 값이 가장 먼저 올수 있도록 정렬을 한다. 이와 같은 일들을 OPEN 리스트가 비워질 때까지 반복하여 최적의 이동경로를 찾는다. 반복이 완료되면 확장된 모든 노드들의 집합인 CLOSE 리스트에서 목표노드로부터 부모노드까지 백트래킹을 이용하여 탐색되어진 최적의 경로를 추출한다.

3.2 동적장애물에 대한 충돌회피 경로설정

동적 장애물 회피를 위해 A*의 공간영역에 대한 탐색문제를 시간공간영역(time-space state domain)의 탐색문제로 확대한다. 현재상태의 노드에 이동하는 장애물들을 포함하여 시공간영역에서 경로탐색을 시행하여 동적인 환경에서 최적의 경로설계를 수행 할 수 있다.

그림 2 는 여러 대의 이동체(로봇, 이동하는 장애물)가 각각의 목표를 향해 최단거리로 이동하는 모습을 보여준다. 실선은 각 로봇들의 이동 방향이고 최단경로를 의미한다. 점선은 각 이동체의 점유 공간이다. 만약 각기 다른 속도를 가진 이동체들이 시간에 대해 고려하

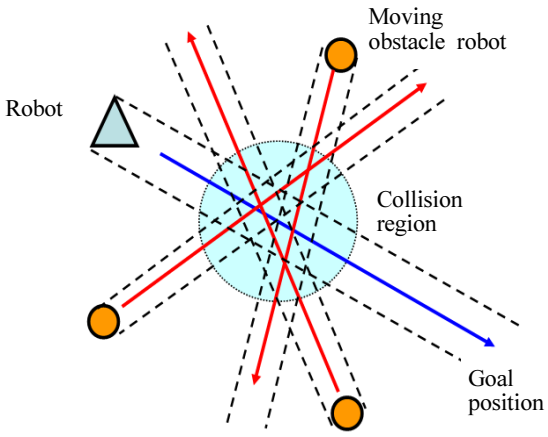


그림 2. 다중 로봇의 이동에 따른 충돌 위험 지역

지 않는다면 충돌이 발생할 가능성은 아주 높아질 것이다. 본 논문에서는 충돌을 방지하기 위해 A* 알고리즘의 공간탐색 문제를 시공간 탐색문제로 탐색공간을 확장하여 속도를 가진 동적인 장애물들을 고려한 최적의 경로계획을 설계하였다. 속도를 가지고 있는 모든 장애물들은 시간의 변화에 따라 이동거리가 달라진다. 이점을 이용하여 기존의 A* 알고리즘 탐색에 속도의 시간변화에 의한 변위를 함께 고려하였다. 그림 3은 이동로봇의 경로계획을 위해 공간영역에서 수행하던 탐색을 시간의 흐름이 변하는 시공간 영역으로 확장한 것을 보여 준다. 먼저 목표를 향해 움직이는 로봇의 이동 가능한 영역을 현재노드를 확장하여 구한다. 이때, 확장할 후계노드들은 현재의 탐색공간인 시간 t에서 구하는 것이 아니라 다음단계의 탐색공간인 $t+\triangle t$시간에서의 탐색공간 중에서 선택한다. 이때 시간 t에서 인접 확장노

드에 없었던 장애물들이 $t+\triangle t$에서는 발생 가능하므로 결국 이동하는 장애물의 위치를 고려한 상태의 확장이 가능하게 된다. 이러한 방식의 확장노드 생성의 반복을 통해 속도를 가진 이동하는 장애물들의 변위를 고려해서 로봇의 최적 경로를 찾아서 시간의 변화에 따라 최적의 경로로 이동할 수 있다.

본 논문에서 제안한 확장된 A*알고리즘을 이용하여 만들어진 알고리즘은 표 2 와 같다. 이 방법의 전체적인 경로계획방식은 A*와 비슷하다. OPEN리스트에 출발노드를 넣고 OPEN 리스트가 완전히 소모될 때까지 반복을 수행한다. 반복 전 초기설정조건에 출발지점 $Start_{pos}$, 목표지점 $Goal_{pos}$과 이동 장애물의 움직임을 계산하기위해 추가적으로 이동하는 장애물들의 속도 o_{vel}와 이동시간 t를 파라미터를 사용한다. 그런 후 A*와 마찬가지로 OPEN 리스트(항상 최소비용으로 순으로 정렬)에서 최적의 노드를 하나씩 꺼내어 목표와 비교하고 이동장애물들의 현재시간 t_{cur}을 식(3)과 같이 갱신한다.

$$t_{cur} = t_{init} + \quad \triangle t^* s_{cs}.step \tag{3}$$

t_{init}은 초기의 설정 시간이고, $\triangle t$는 시간 간격, $s_{cs}.step$은 현재상태 노드의 스텝값이다. $\triangle t^* s_{cs}.step$은 알고리즘 반복의 각 단계에서 변화하는 시간이다. 갱신된 시간을 바탕으로 이동하는 장애물들의 위치도 마찬가지로 갱신한다. 로봇의 위치를 계산할 때 시간은 $\triangle t$이고

표 2. 시간축으로 확장된 A* 알고리즘

Find_Path($Goal_{pos}$, $Start_{pos}$, t, o_{vel})

```
// Initialize_Queue(Q_open, Q_dose);
// Initialize_Value();
s_init ← Start_pos, Goal_pos, t, o_vel ;
Q_open.insert(s_init);
Moveobstacle(t_init, o_vel) ;
while Q_open != 0 do
  s_cs ← Q_open.pop( );
  Q_dose. insert(s_cs) ;
  if GoalReach( s_cs ) then
     Return robot_path;
  end
  t_cur = t_init + △t* s_cs.step;
  Moveobstacle(t_cur, o_vel) ;
  foreach s_suc ∈ s_all suc do
    if Not SafeSuccessor then
      Discard S_suc ;
    end
    s_suc ← Q_open.pop( ) ;
    c_l ← LocationCost( s_suc ) ;
    c_h ← HeuristicCost(s_suc, Goal_pos) ;
    Q_open.insert(s_suc, s_suc. cost+c_l+c_h,t_cur) ;
  end
  Q_open. Sort();
end
```

장애물들의 위치를 계산할 때 시간은 $\triangle t^* s_{cs}.step$이 된다. 변화된 장애물들의 위치를 맵에 반영하여 후계노드의 안전성을 검사해 불안전하면 다음번 후계노드를 검사하여 안전성이 확보될 때까지 반복한다. 안전한 후계노드가 나타나면 그 노드에 대한 식(2)의 평가함수 값을 구하고 각 리스트들의 중복성을 검사한 후 OPEN 리스트에 넣는다. OPEN 리스트가 남아있지 않을 때까지 반복하여 최적의 이동경로를 획득한다. 제안된 알고리즘에서 얻을 수 있는 최적해는 적격성을 유지하기 위해서는 식(4)를 만족해야한다.

$$\hat{h}(n) \leq h(n) \tag{4}$$

$h(n)$은 최적의 경로를 얻기 위한 최적의 평가값을 의미한다. 추정된 잔여거리인 휴리스틱 평가값 $\hat{h}(n)$이 최적의 평가값 $h(n)$보다 작거나 같으면 최적의 해를 만족할 시킬 수 있다. 즉, $\hat{h}(n)$이 노드 n에서 목적점 $Goal_{pos}$까지의 실제 최단거리보다 크지 않다면 최단경로를 얻는 것이 가능하다.

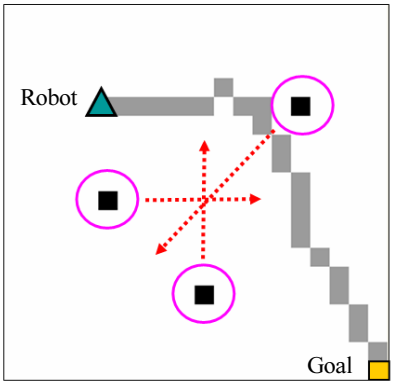
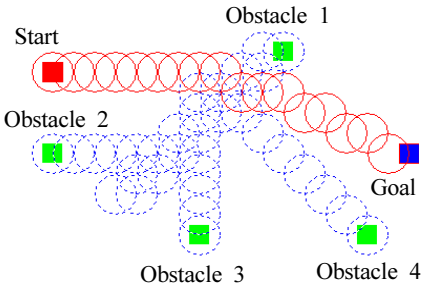


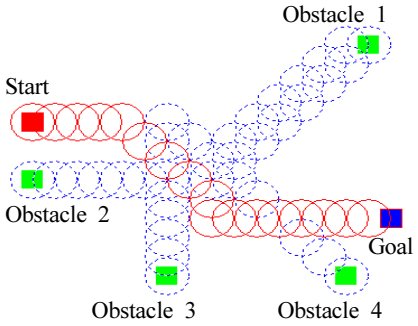
그림 4. 3대의 이동 장애물에 대한 충돌회피탐색

4. 실 험

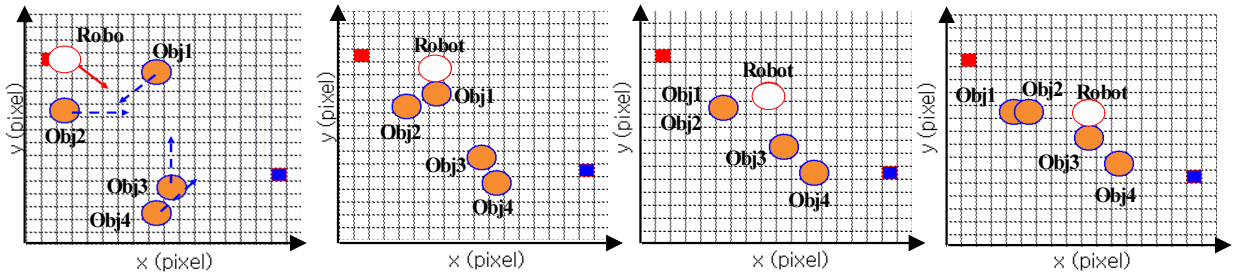
본 논문에서 제안된 방법인 확장된 A*알고리즘을 이용하여 시간영역에서 선형 속도와 방향을 가진 동적인 장애물에 대해 모의 실험을 수행하였다. 동적 장애물을 가진 환경에서 공간영역뿐만 아니라 시간영역에서 목표를 향해 경로를 설계할 때 장애물들은 일정한 선형속도를 유지하도록 하였고, 방향은 임의로 주었다. 선형속도를 가지고 움직이는 장애물들은 8 방향의 방위를 이용하였다. 그림 4 는 충돌회피 시뮬레이션 결과로 로봇이 목표 지점까지 경로를 탐색할 때 검은색 사각형인 장애물이 점선으로된 화살표방향으로 이동할 때 로봇과의 충돌을 회피하는 것을 표현한다. 원의 크기는 이동장애물의 크기를 표시한다. 실험은 가상의 선형속도를 가진 이동하는 복수의 장애물을 설정하고 이동로봇이 시간에 따라 움직이는 장애물들과의 충돌을 회피하는 이동경로를 관측하였다. 실험에 사용한 환경은 데스크탑 PC에서 윈도우XP OS하에서 수행되었다. 그림 5 는 제안한 알고리즘을 이용하여 동적인 장애물들이 있는 환경에서 시뮬레이션하여 얻어진 로봇과 장애물들의 궤적들과 시간축에 대한 각 이동장애물과 로봇의 접촉거리이다. 빨간색 사각형은 이동로봇의 출발위치이고 파란색 사각형은 도달하는 목표 위치이다. 녹색 사각형은 이동하는 장애물들의 초기위치이다. 빨간색 실선의 원들은 이동로봇의 궤적들이고, 파란색 점선으로 표시된 원은 장애물들의 궤적이다. 장애물들은 서로 다른 방향으로 이동한다.



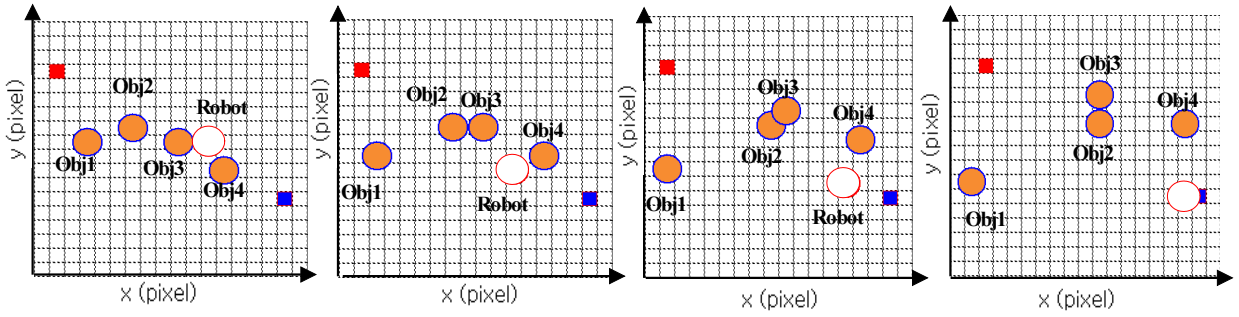
(a)



(b)

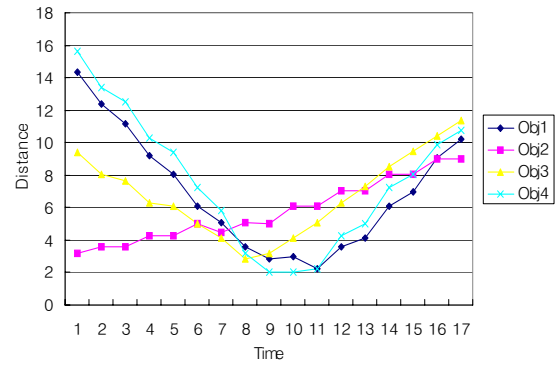


(a)

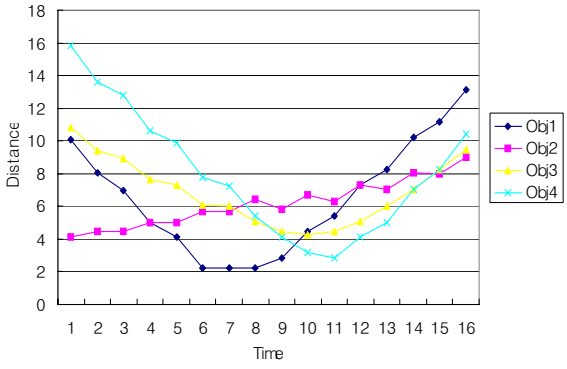


(b)

그림 6. 이동 로봇의 경로 이동 시뮬레이션 캡쳐 영상



(c)



(d)

그림 5. 이동 로봇과 움직이는 장애물들의 궤적 및 이동로봇과 장애물들의 상대거리

장애물 1 은 좌측하단, 장애물 2 는 우측, 장애물 3 은 상단, 장애물 4 는 좌측상단 방향으로 진행한다. 초기 장애물 방향과 속도는 설정가능하고 속도는 일정하게 진행한다. 그림 5 (a)와(b)는 출발점과 목표 점은 같지만 장애물들의 위치에 따라 이동로봇이 움직인 궤적이 바뀐 모습을 관측할 수 있다. 그림 5 (c)와 (d)는 각각(a)와(b)에 대한 로봇과 이동장애물의 거리를 시간축으로 나타낸 그림이다. 거리는 로봇과 이동장애물 중심간의 거리이고 단위는 pixel 이다. 이동체의 반경은 1 단위pixel이므로 로봇과 이동장애물과의 최소 거리는 2 단위 pixel이 된다. 그림 5 (c)와 (d)에서 가장 근접했을 때의 거리가 2 단위 pixel인 것을 확인할 수 있다. 그림 6 은 이동로봇과 움직이는 장애물의 컴퓨터 시뮬레이션 데모 캡쳐 사진이다. 출발과 목표지점의 표현은 그림 5 와 같다. 이동로봇은 흰색 원으로 표시했고, 움직이는 장애물들은 주황색 원으로 표현했다. 각 그림들은 이동로봇이 움직이는 장애물들의 이동경로를 피해 최적의 경로를 탐색해 목표지점을 찾아가는 모습을 보여준다. 원의 크기는 물체의 반경을 의미한다. 이동하는 물체와 부딪히지 않으면서 최적 경로를 탐색하여 찾아가는 것을 볼 수 있다.

5. 결 론

동적인 환경에서 움직이는 장애물에 대한 충돌회피연구는 최근 활발히 진행 중인 분야이다. 본 논문에서는 복수의 이동장애물에 대해 공간영역 뿐만 아니라 이동하는 장애물에 의해 점유되는 시간-공간 영역을 고려하여 경로를 탐색해 감으로써 동적인 환경에 민첩하게 대응한다. 논문에서는 프로그램 속도를 높이기 위해 속도와 방향을 선형적으로 제한했다. 추후 연구 방향으로 비선형 속도 및 자율 방향에 대해 고려하여 더욱 일반적인 동적 환경에 대한 접근방법을 만드는 것이 필요하다. 향후 현재의 연구를 기반으로 하여 장애물들이 이동하는 동안에 속도와 방향의 변화를 고려한 최적의 경로 재탐색 알고리즘에 대해 연구를 진행하고 이를 기반으로 이동로봇에 탑재하여 실세계에서 실험을 수행할 것이다. 제안한 알고리즘은 최근 활발히 연구 중인 휴머노이드 로봇의 풋스텝 경로계획에까지 확장해서 적용할 수 있다. 뿐만 아니라 경로계획이 사용되는 다양한 분야에서 적용가능하리라 본다.

참 고 문 헌

[1] J. Chestnutt, M. Lau, G. Cheung, J. Kuffer, J. Hodgins, and T. Kanade, “Footstep Planning for the Honda ASIMO Humanoid”, *IEEE International Conference on Robotics and Automation*, pp.629-634, 2005.

[2] S. Koenig and M. Likhachev, “*Incremental A*,*” In Advances in Neural Information Processing Systems 14, MIT Press, 2002.

[3] P. O'Donnell and T. Lozano-Perez. “Deadlock-free and collision-free coordination of two robot manipulators.” *IEEE International Conference on Robotics and Automation*, pp.484-489, 1989.

[4] C. Stachniss and W. Burgard, “An integrated approach to goaldirected obstacle avoidance under dynamic constraints for dynamic environments,” *In Conference on Robotics and Automation*, 1994.

[5] T. Fraichaed and A. Scheuer, “Car-like robots and moving obstacles,” *IEEE International Conference on Robotics and Automation*, pp.64-69 1994.

[6] M. Seder, K. Macek, and I. Petrovic, “An integrated approach to realtime mobile robot control in partially known indoor environments,” *In Proc. of the 31st Annual Conference of the IEEE Industrial Electronics Society*, pp.1785-1790, 2005.

[7] C. Stachniss and W. Burgard, “An integrated approach to goaldirected obstacle avoidance under dynamic constraints for dynamic environments,” *IEEE International Conference on Intelligent Robots and Systems*, 2002.

[8] D. Fox, W. Burgard, and S. Thrun, “The dynamic window approach to collision avoidance,” *IEEE Robotics and Automation Magazine*, vol.4, no.1, pp.23-33, 1997.

[9] R. Simmons, “The curvature-velocity method for local obstacle avoidance,” *IEEE International Conference on Robotics and Automation*, vol.4, pp.3375-3382, 1996.

[10] Y. K. Nak and R. Simmons, “The lane curvat- ure method for local obstacle avoidance,” *IEEE/RSJ International Conference on Robotics and Automation*, vol.3, pp.1615-1621, 1998.

[11] P. Fiorini and Z. Shiller, “Motion planning in dynamic environments using velocity obstacles,” *International Journal of Robotics Research*, no.17, pp. 711-727, 1998.

[12] F. Large, “Navigation autonome d’'un robot mobile en environnement dynamique et incertain,” Ph.D. diss., University de Savoie, 2003.

[13] T. Fraichard and H. Asama, “Inevitable collision states a step towards safer robots?” *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp.388-393, 2003.

[14] E. Owen and L. Montano, “A robocentric motion planner for dynamic environments using the velocity space,” *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp.4368-4374, 2006.

[15] C. Cou´e, “Mod`ele bay´esien pour l’'analyse multimodale d’'environnements dynamiques et encombr´es: application`a l’'assistance`a la consuite automobile en milieu urbain.” Ph.D. diss., Inst. Nat. Polytechnique de Grenoble, 2003.

[16] C. Fulgenzi, A. Spalanzani, and C. Laugier, “Dynamic Obstacle Avoidance in uncertain environment combining PVOs and Occupancy Grid”, *IEEE International Conference on Robotics and Automation*. pp.1610-1616, 2007.

[17] C. Cou´e, C. Pradalier, C. Laugier, T. Fraichard, and P. Bessi`ere, “Bayesian occupancy filtering for multitarget tracking: an automotive application,” *International Journal of Robotics Research*, vol. 25, no.1, pp.19-30, 2006.

김 대 광

2006년 동명대 컴퓨터 공학과 졸업.

2006년 ~ 현재 부산대 지능기계과 석사과정

관심분야 : 컴퓨터 비전, 지능형 로봇, 패턴 인식

강 동 중

1999년 KAIST 자동화 및 설계 공학과 공학박사

1990~1992년 현대전자 산전 연구소 연구원

1997~1999년 삼성종합기술원 신호처리연구실 선임 연구원

2004년 미국 Cornell Univ. 방문연구원

2000년~2005년 동명대학교 메카트로닉스 공학과 조교수

2006~현재 부산대 기계공학부 조교수

관심분야 : 컴퓨터 비전, 이동로봇, 영상기반 검사 시스템 개발