

격자화된 공간상에서 4중-나무 구조를 이용한 가시성 검사를 바탕으로 한 새로운 경로 계획 알고리즘과 그 개선 방안들

New Path Planning Algorithm based on the Visibility Checking using a Quad-tree on a Quantized Space, and its improvements

김 정 태*, 김 대 진
(Jungtae Kim and Daijin Kim)

Abstract: In this paper, we introduce a new path planning algorithm which combines the merits of a visibility graph algorithm and an adaptive cell decomposition. We quantize a given map with empty cells, blocked cells, and mixed cells, then find the optimal path on the quantized map using a visibility graph algorithm. For reducing the number of the quantized cells we use the quad-tree technique which is used in an adaptive cell decomposition, and for improving the performance of the visibility checking in making a visibility graph we propose a new visibility checking method which uses the property of the quad-tree instead of the well-known rotational sweep-line algorithm. For the more efficient visibility checking, we propose two additional improvements for our suggested method. Both of them are used for reducing the visited cells in the quad-tree. The experiments for a performance comparison of our algorithm with other well-known algorithms show that our proposed method is superior to others.

Keywords: path-planning, visibility graph, adaptive cell decomposition, quad-tree

I. 서론

경로 계획 문제는 주어진 공간상의 어느 시작점으로부터 도착점까지 어떠한 장애물과의 충돌 없이 안전하게 이동하는 방법을 찾는 문제로 로봇 분야를 비롯하여, 컴퓨터 그래픽, 인공지능 게임, 생명공학, 가상 환경, 프로토타입 제작, 수술 시뮬, CAD 시스템 등 여러 분야에서 다양하게 다루고 있다. 특히 이동 로봇 분야의 경우 로봇의 이동 문제는 경로 계획 문제와 직결되기 때문에 오랜 기간에 걸쳐 다양한 연구가 진행되어 오고 있다[1-4].

이동 로봇에 부착된 근접 센서를 이용하여 목적지까지 도달하기 위한 방법으로 제안된 것으로, 버그 알고리즘(bug algorithms) [5-7]과 그의 변종들, 그리고 벡터 필드 히스토그램(vector field histogram) 방법[8] 등이 있다. 이 방법들은 모두 목적지까지 이동하는 도중에 장애물을 만났을 때 일정한 규칙에 따라 동작하게 함으로서 최종적으로 안전하게 목적지까지 도달하도록 하는 방법이다. 하지만 장애물의 배치나 전체 지도의 모양을 고려하지 않기 때문에, 동작 알고리즘은 단순하나 체계적이고 효율적인 경로를 찾는다고는 보장할 수 없다.

이와는 달리 전체 지도의 모양을 고려하여 경로를 정하고 그에 맞추어 이동하는 알고리즘도 있다. 보로노이 다이어그램(Voronoi diagram)을 이용한 방법[9-13], 잠재 영역 함수(potential field function)를 이용한 방법[14-18], 확률적인 샘플링 방법을 이용하는 방법들[19,20], 그리고 다음 장의 선행 연구

에서 자세히 다룬 가시성 그래프(visibility graph)를 이용한 방법[21,22]과 셀-분할(cell decomposition) 알고리즘을 이용한 방법[1,3] 등 다양한 알고리즘이 존재한다.

본 논문은 가시성 그래프와 셀-분할 알고리즘의 하나인 가변-셀-분할 알고리즘을 결합한 새로운 경로 계획을 제시한다. II 장에서 관련 연구의 소개를 하고, III 장에서 두 알고리즘을 결합한 새로운 알고리즘을 소개한다. IV 장에서는 제시한 알고리즘의 성능 향상을 위해 개선할 수 있는 방안을 소개한다. 그리고 실험과 결과 분석을 통해 제시한 알고리즘의 타당성을 보이고, 그 결과를 정리하는 것으로 논문을 마무리 짓는다.

II. 관련 연구

대부분의 경로 계획 알고리즘[2,3]은 크게 두 단계로 구분되어 동작한다. 하나는 시작점과 도착점을 포함하는 빈 공간상의 그래프를 만드는 단계와, 다른 하나는 만들어진 그래프에서 최단 경로를 찾는 단계이다. 그래프의 최단거리를 찾기 위해 넓이-우선 탐색(breadth-first search) 방법, 깊이-우선 탐색(depth-first search), 다익스트라 알고리즘(Dijkstra's algorithm)[23], 등의 그래프 찾기 알고리즘(graph search algorithm)을 이용한다.

빈 공간 상에 그래프를 만드는 방법 가운데 가장 잘 알려진 방법인 가시성 그래프(visibility graph)와 몇 가지 셀-분할(cell decomposition) 알고리즘에 대해 알아보도록 하겠다.

1. 가시성 그래프 알고리즘

모든 장애물의 꼭지점을 연결한 선들 가운데서 최단 경로를 찾아, 이 경로를 통해 출발지에서 도착지까지 이동하는 방법이 가시성 그래프 알고리즘이다(그림 1(a)). 가시성 그래프 알고리즘은 2차원 공간상에서 장애물들이 다각형을 띠고 있다는 가정하에서는 최단의 경로를 찾아주는 알고리즘이다. 이 가정이 성립하지 않는 일반적인 상황에서는 공간의 격자화와 같은 방법을 통하여 장애물을 다각형의 구조물로 변환시키는 선행 작업을 하게 된다.

* 책임저자(Corresponding Author)

논문접수: 2009. 10. 20., 수정: 2009. 11. 2., 채택확정: 2009. 11. 2.

김정태, 김대진: 포항공과대학교 컴퓨터공학과, 생체인식연구센터 (postman@postech.ac.kr/dkim@postech.ac.kr)

※ 본 연구는 (포항공과대학교 고품질 로봇비전 연구센터를 통한) 지식경제부/한국산업기술진흥원 융복합형 로봇 전문 인력 양성 사업의 지원과 2002년도 정부(교육과학기술부)의 재원으로 한국연구재단 지정 생체인식연구센터의 지원을 받아 이루어졌음. (2009-0062991)

※ 상기 논문은 KACC2009 학술대회에서 초안이 발표되었습니다.

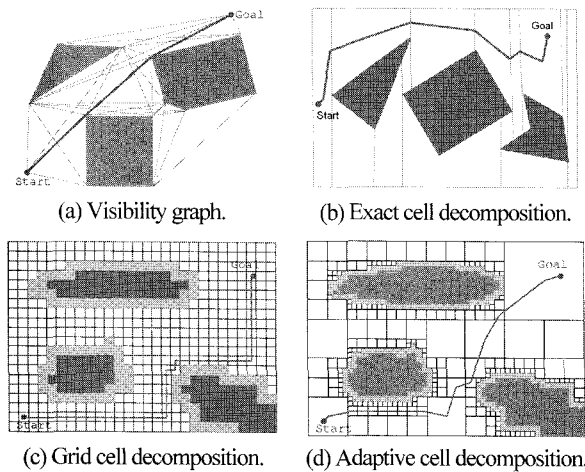


그림 1. 다양한 경로 계획 알고리즘.
Fig. 1. Path planning algorithms.

2. 셀-분할 알고리즘

셀-분할 알고리즘은 주어진 공간에서 장애물을 제외한 빈 공간을 일정한 방법으로 세분화하여, 각 공간의 중심을 이웃하는 공간의 중심과 연결하는 그래프를 구성하여, 이 그래프 상에서의 최단 경로를 구하는 방법이다. 공간을 나누는 방법에 따라 셀-분할 알고리즘이 세분화 된다.

정확한-셀-분할 알고리즘(ECD: Exact Cell Decomposition)은 다각형 장애물의 각 꼭지점을 지나 상하단의 경계나 다른 장애물을 만나는 지점까지 수직의 선을 그어서 장애물이 아닌 공간을 나누는 방법이다(그림 1(b)). 각각의 통로가 주위의 장애물로부터 최대한 멀리 떨어져 있어서 안전한 이동 경로를 찾아준다. 하지만 이 방법은 다각형 장애물이 존재하는 경우에만 적용이 가능하다.

격자-셀-분할 알고리즘(GCD: Grid Cell Decomposition)은 전체 공간을 일정한 크기의 격자로 나눈 다음, 세분화된 공간을 장애물 영역, 빈 공간 영역, 그리고 혼합된 영역으로 나누어서 빈 공간 영역들간에 그래프를 구성하는 방법이다(그림 1(c)). 격자 크기를 너무 작게 하면 각 공간의 정보를 담기 위한 메모리 공간이 많이 필요하게 되고, 격자 크기를 너무 크게 하면 장애물 사이의 연결이 가능한 통로가 막혀 버리는 문제가 생길 수 있다.

가변-셀-분할 알고리즘(ACD: Adaptive Cell Decomposition)은 GCD에 의해 세밀하게 나누어진 격자 공간에서 이웃하는 네 영역이 같은 성질의 영역일 때 하나로 묶기를 반복하는 방식으로 영역의 개수를 줄인 다음, 빈 공간 영역들간에 그래프를 구성한다(그림 1(d)). GCD에 비해 영역의 개수가 훨씬 적어지기 때문에 보다 효과적으로 전체 공간을 표현할 수 있다.

III. 다중 단계 격자화 가시성 그래프(MQVG)

앞서의 선행 연구 장에서 설명한 가시성 그래프와 가변-셀-분할 알고리즘의 장점을 모으고 단점을 보완한 새로운 알고리즘을 본 연구에서 제시하고자 한다. 제시하는 알고리즘의 세부 과정들을 다음의 항에서 간략히 소개하겠다. 좀 더 자세한 설명은 앞서 발표된 논문[24,25]을 참조하기 바란다.

1. 격자화

공간의 격자화는 격자-셀-분할 알고리즘에서 사용된 격자

화 방법과 동일하다. 전체 공간을 일정한 크기의 격자로 나눈 다음, 각 영역을 다음의 2가지 영역 가운데 하나로 표시한다. 1) 막힌 공간 : 전체나 일부가 장애물로 채워진 영역, 2) 빈 공간 : 전체 영역에 장애물이 존재하지 않는 영역.

2. 격자 공간의 4중-나무 구조화

격자화 작업을 통하여 구분되어지는 각 영역 가운데 막힌 공간에 대해서, 이웃하는 네 공간이 모두 동일할 경우 하나의 막힌 공간으로 통합하여 표시하는 방법으로 4중-나무 구조(quad-tree)를 구성할 수 있다. 이는 가변-셀-분할 알고리즘과 비슷하지만, 빈 공간 대신에 막힌 공간을 대상으로 4중-나무 구조를 구성한다는 점에서 차이가 있다.

3. 꼭지점 찾기

가시성 그래프 알고리즘을 적용하기 위하여 격자화된 장애물의 꼭지점을 찾는다. 막힌 공간의 네 모서리 지점에 각각 점을 찍어서 같은 지점에 몇 번의 점이 찍혔는지 수를 센다. 그러면, 점이 한 번 찍힌 지점은 튀어 나온 모서리가 되고, 세 번 찍힌 지점은 움푹 들어간 모서리가 된다. 움푹 들어간 모서리는 가시성 그래프의 최종 결과에 포함되지 않기 때문에[2], 한번 점이 찍힌 지점만이 최후의 꼭지점이 된다.

4. 가시성 확인

꼭지점들 가운데 임의의 두 꼭지점을 잇는 선이 중간에 장애물의 방해 없이 연결이 가능한지 가시성 여부를 판별한다. 이를 위하여 기존의 가시성 그래프에서는 RSL (Rotational Sweep-Line) 알고리즘 [26]을 이용하지만, MQVG에서는 격자 공간의 4중 나무 구조를 이용하여 꼭지점들 가운데 임의의 두 개를 선택하여 이 둘을 잇는 선과 모든 막힌 공간을 비교하여 하나라도 교차할 경우 가시성이 없다고 판별하는 방법을 이용하였다.

IV. 알고리즘의 성능 개선

본 장에서는 앞서 소개된 경로 계획 알고리즘의 성능 향상을 위한 두 가지 방법을 소개하고자 한다. 제시된 알고리즘에서 임의의 두 점간의 가시성 여부를 판단하는 부분이 전체 알고리즘 가운데 가장 시간을 많이 소모한다. 따라서 가시성 판단을 보다 빠르게 하기 위한 방법을 다음 절에 걸쳐 소개한다.

1. 빈 공간 영역에 대한 4중 나무 구조화

앞서 소개된 알고리즘의 4중 나무 구조화 부분을 보면 막힌 공간에 대해서만 통합 작업을 수행하였다. 이것은 뒤에 막힌 공간에 대해서만 가시성 검사를 하기 때문이다. 하지만 전체 트리에서 막힌 공간만을 찾기 위해서는 모든 뿌리-노드(root node)에서 잎새-노드(leaf node)까지를 검사하는 과정에서 빈 공간을 나타내는 노드를 확인하는 불필요한 시간이 소모된다. 이 같은 불필요한 확인 과정을 최소화 시키기 위하여 빈 공간을 나타내는 노드에 대해서도 통합 과정을 수행한다. 그러면 빈 공간을 나타내는 노드들의 확인 작업이 줄어들게 되어 전체 속도의 향상된다.

그림 2의 왼쪽 4중 나무 구조는 기존에 막힌 공간을 나타내는 노드에 대해서만 통합 작업을 한 구조이며, 오른쪽 4중 나무 구조는 빈 공간을 나타내는 노드에 대해서도 같은 통합 작업을 하여 얻어진 구조를 나타낸다. 그림에서 보는 바와

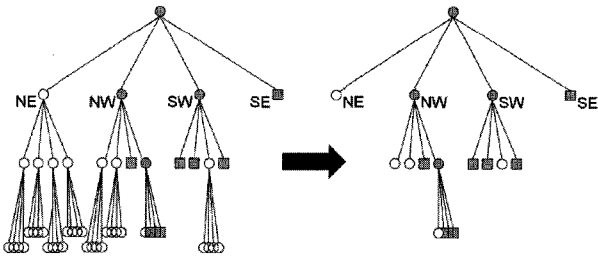


그림 2. 빈 공간 노드의 최소화로 속도를 향상 시킴. 흰색 원은 빈 공간을 나타내고, 회색 사각형은 막힌 공간, 회색 원은 혼합 공간을 나타냄.

Fig. 2. Speed up by reducing the empty nodes. White circles indicate empty cells, gray squares indicate blocked cells, and gray circles indicates the mixture of them.

같이 보다 적은 수의 잎새-노드만 남게 되는 것을 확인할 수 있다. 적은 잎새-노드의 수는 가시성 확인 시간의 단축으로 이어진다는 것을 뒤에 실험 결과를 통해 확인할 수 있다.

2. 연결성 각도를 이용한 가시성 확인 속도 향상

4중 나무 구조의 노드 접근을 최소화하는 것은 가시성 확인 작업의 속도 향상과 전체 알고리즘의 속도 향상과 직결된다. 따라서 최대한 4중 나무 구조의 노드 검사 횟수를 줄이는 것이 중요하다.

이 절에서는 다음의 세 단계를 거친 성능 향상 방법을 소개한다. 우선, 가시성 검사를 하고자 하는 두 지점의 위치를 p1, p2라고 한다. 이때 p1과 p2가 속해 있는 막힌 잎새-노드의 정보도 같이 얻을 수 있다. 왜냐하면 가시성 그래프의 한 지점은 막힌 잎새-노드의 한 모서리에서 얻기 때문이다. 이렇게 얻은 각각의 잎새-노드를 c1과 c2라고 할 때, c1과 c2의 상위(부모) 노드를 추적하여 공통되는 노드를 찾는다. 이 노드를 c0라고 하자. 그러면, p1과 p2의 가시성 검사는 c0 노드에 해당하는 하위 막힌 영역에 대해서만 검사를 하는 것으로 충분하게 된다(그림 3).

그 다음으로 c0 노드에서 하위 노드들에 대해 다음과 같은 검사를 한다. 우선 빈 공간을 나타내는 노드는 제외한다. 그리고 막힌 영역을 나타내는 공간 가운데 하나를 c3이라고 하였을 때, 다음의 조건을 만족하는 경우 이 영역에 포함되는 모든 노드를 제외한다.

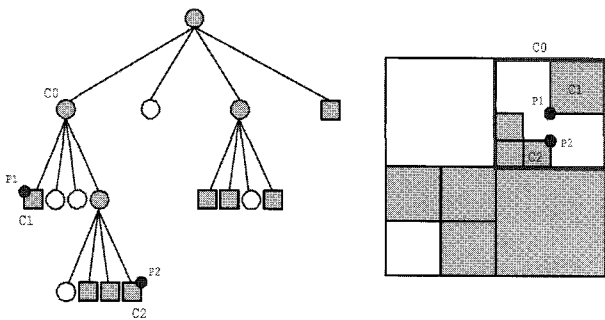


그림 3. 4중 나무 구조의 공통된 노드를 찾아, 이에 대해서만 가시성 검사를 함.

Fig. 3. Find the common node of a node containing p1 and a node containing p2, check the visibility for them.

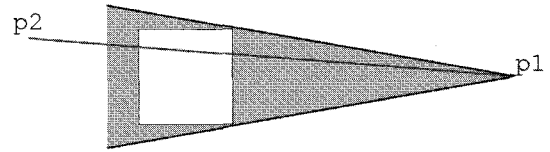


그림 4. 비교 영역의 각도 정보를 이용한 가시성 확인 작업.
Fig. 4. Speed up by calculating the angle of p1 and p2, and comparing the four angles of the cell from p1.

$$\begin{aligned}
 &1) \max(p1.x, p2.x) < \min_x(c3) \\
 &2) \min(p1.x, p2.x) > \max_x(c3) \\
 &3) \max(p1.y, p2.y) < \min_y(c3) \\
 &4) \min(p1.y, p2.y) > \max_y(c3)
 \end{aligned} \tag{1}$$

마지막으로 남은 막힌 공간들(c4)에 대해, p1에서 사각 영역의 각 꼭지점들과의 각도를 계산하여 최대각을 max_ang(c4), 최소각을 min_ang(c4), p1과 p2를 잇는 선의 각을 ang(p1,p2)라고 하였을 때, 다음 조건을 만족하는 노드가 있을 때, p1과 p2 사이에 가시성이 없다고 판별한다(그림 4).

$$\min_ang(c4) < ang(p1,p2) < \max_ang(c4) \tag{2}$$

위의 선행 조건을 만족하는 막힌 공간 노드에 대해서 기존의 가시성 확인 작업을 하여, 최종적으로 가시성 여부를 판단하게 된다.

이런 선행 작업은 연결선의 가시성 여부 확인과 별도로 이루어지기 때문에 그 만큼의 추가 시간이 소모된다. 하지만 다음의 실험 결과에서 보듯이, 추가 작업을 통하여 줄어들게 되는 가시성 여부 확인 작업이 더 많기 때문에, 전체적으로는 알고리즘의 속도 향상이 일어나게 된다. 자세한 성능 향상 수치는 다음의 실험 및 결과 장에서 보여진다.

V. 실험 및 결과

본 논문에서는 경로 계획 알고리즘으로 잘 알려진 가시성 그래프와 가변-셀-분할 알고리즘의 장단점을 보완한 새로운 알고리즘을 제시하였다. 또한 새로운 알고리즘의 성능 향상을 위하여 두 가지 성능 향상 방법을 추가로 소개하였다. 처음 제시된 알고리즘을 MQVG라고 명하고, 빈 공간 영역에 대해서도 4중 나무 구조를 적용한 방법을 MQVG v1이라 하자. 그리고 MQVG v1에 연결성 각도 정보를 이용하여 성능을 향상시킨 알고리즘을 MQVG v2라고 하자. 소개된 알고리즘의 성능 평가를 위하여 그림 5의 세 가지 샘플 공간을 사용하였다. 각 공간은 1024x1024의 지도로 검은색이 장애물에 해당되고 시작점은 좌측 하향에, 도착점을 우측상향에 잡았다. 모든 실험은 Intel(R) Core(TM)2 Quad CPU 2.66 GHz에 8GB RAM인 컴퓨터에서 테스트하였다.

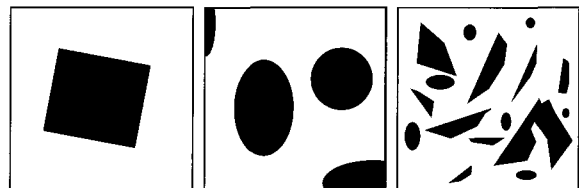


그림 5. 샘플 공간. 검은색이 장애물.
Fig. 5. Sample spaces. Black regions represent obstacles.

표 1. 알고리즘 시간(초) 비교 (? : 시간 초과).

Table 1. Time comparison among algorithms.

공간 수	공간	GCD	ACD	MQVG
$2^5 \times 2^5$	1	33.38	1.28	0.17
	2	20.63	1.84	0.17
	3	18.02	3.01	0.62
$2^6 \times 2^6$	1	1448.32	3.32	0.21
	2	1115.59	15.91	0.53
	3	1139.20	34.46	5.78
$2^7 \times 2^7$	1	?	56.45	0.64
	2	?	155.00	4.54
	3	?	501.18	56.40

표 2. 가시성 확인을 위한 평균 노드 방문 수.

Table 2. Average visits for visibility checking.

공간 수	공간	MQVG	MQVG v1	MQVG v2
$2^5 \times 2^5$	1	44	44	3
	2	96	96	5
	3	102	102	6
$2^6 \times 2^6$	1	68	68	4
	2	193	193	7
	3	219	219	7
$2^7 \times 2^7$	1	180	180	6
	2	380	380	8
	3	556	556	10

표 3. 알고리즘 시간(초) 비교.

Table 3. Time comparison among algorithms.

공간 수	공간	MQVG	MQVG v1	MQVG v2
$2^5 \times 2^5$	1	0.17	0.16	0.16
	2	0.17	0.17	0.15
	3	0.62	0.57	0.46
$2^6 \times 2^6$	1	0.21	0.19	0.19
	2	0.53	0.44	0.30
	3	5.78	4.21	2.33
$2^7 \times 2^7$	1	0.64	0.33	0.27
	2	4.54	2.32	1.12
	3	56.40	30.57	11.37

우선 이 논문에서 제시한 MQVG 알고리즘과 기존의 격자-셀-분할 알고리즘(GCD), 가변-셀-분할 알고리즘(ACD)의 성능 비교를 위하여, 세 개의 샘플 공간에 대해 세 가지 격자 최소 공간 크기를 달리 설정하여 실험하였다. 성능 평가는 경로 계획을 알아내기 위해 걸린 시간 비교로 하였다(표 1).

실험 결과, 공간 구조에 따라 조금씩 성능 차이가 존재하지만, 대체적으로 가장 작은 셀의 크기가 작을 수록 시간이 증가하는 것을 볼 수 있다. 모든 경우에 GCD 보다 ACD의 성능이 좋았고, 제시한 MQVG 알고리즘이 가장 빨리 시작점에서 도착점까지의 경로를 찾았다.

다음 실험은 제시된 알고리즘의 성능 향상을 위하여 추가로 제안한 방법이 얼마만큼의 성능 향상을 가져오는지를 확인해 보았다. 표 2는 알고리즘 개선으로 인해 임의의 노드들 간의 가시성 검사를 위해 필요한 평균 영역 방문 횟수를 정리하였으며, 표 3은 알고리즘의 총 연산 시간을 비교하였다.

표 2에서 보면, MQVG v1의 경우에는 평균 노드 방문 수가

MQVG와 차이가 없지만, MQVG v2의 경우에는 그 수가 급격히 줄어들었음을 확인할 수 있었다. 몇 번의 노드 확인으로 임의의 두 점 사이의 가시성 여부가 판단될 수 있음을 보여주고 있다. MQVG v1의 추가 작업이 알고리즘의 성능 향상에 도움이 없는 것 같지만, 표 3에서 보면 전체 알고리즘의 속도 개선에는 기여를 한 것을 볼 수 있다. 모든 실험에서 MQVG v1의 경우가 속도 개선을 가져왔고, MQVG v2의 경우에는 50% 이상의 시간을 단축시킨 것을 확인할 수 있다.

VI. 결론

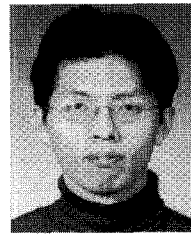
본 논문은 기존의 경로 계획에서 잘 알려진 가시성 그래프와 가변-셀-분할 알고리즘을 장단점을 융합, 보완한 새로운 경로 계획 알고리즘을 제안한다. 가변-셀-분할 알고리즘을 통해 찾은 경로는 시작점에서 도착점까지의 최단 경로가 아니기 때문에 여기에 가시성 그래프 방법을 결합하여 격자 공간 상에서 최단의 경로를 구할 수 있도록 하였다. 그리고 이때 사용한 4중 나무 구조를 이용하여 기존의 가시성 그래프가 사용하는 RSL 알고리즘을 사용하지 않고도 가시성 여부를 판별할 수 있는 방법을 제안하였다. 이렇게 제안된 알고리즘의 성능 향상을 위하여 추가도 2가지 개선 방안을 더 소개하였다. 실험을 통하여 우리가 제시한 알고리즘이 기존의 방법에 비해 성능이 좋음을 확인하였다.

참고문헌

- [1] H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun, *Principles of Robot Motion: Theory, Algorithms and Implementations*. Cambridge, MA, USA: MIT Press, 2005.
- [2] J. C. Latombe, *Robot Motion Planning*, Kluwer Academic, Boston, 1991.
- [3] R. Siegwart and I. R. Nourbakhsh, *Introduction to autonomous mobile robot*, Cambridge, MA, USA: MIT Press, 2004.
- [4] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*, Cambridge, MA, USA: MIT Press, 2005.
- [5] V. J. Lumelsky and A. A. Stepanov, "Path planning strategies for a point mobile automaton moving amidst unknown obstacles of arbitrary shape," *Algorithmica*, 2:403-430, 1987.
- [6] I. Kamon and E. Rivlin, "Sensory-based motion planning with global proofs," *IEEE Transactions on Robotics and Automation*, 13(6):814-822, Dec. 1997.
- [7] S. L. Laubach and J. W. Burdick, "RoverBug: Long range navigation for mars rovers," *Lecture Notes in Control and Information Sciences*, pp. 339-348, 1999.
- [8] J. Borenstein and Y. Koren, "The vector field histogram - fast obstacle avoidance for mobile robots," *IEEE Journal of Robotics and Automation*, vol. 7, no. 3, pp. 279-288, Jun. 1991.
- [9] R. V. Benson, *Euclidean Geometry and Convexity*. New York: McGraw-Hill, 1966.
- [10] H. Choset, "Incremental construction of the generalized Voronoi diagram," in *Proc. of the 1st CGC Workshop on Computational Geometry*, Oct. 1997.
- [11] L. Guibas, L. Ramshaw, and J. Stolfi, "A kinetic framework for computational geometry," *Foundations of Computer Science*, 1983., 24th Annual Symposium on, pp. 100-111, Nov. 1983.
- [12] H. Choset, "Incremental construction of the generalized Voronoi

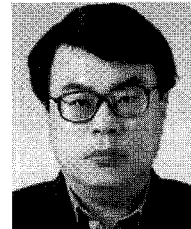
- diagram, the generalized Voronoi graph, and the hierarchical generalized Voronoi graph," in *Proc. Of the 1st CGC Workshop on Computational Geometry*, Oct. 1997.
- [13] H. Choset, S. Walker, K. Eiamsa-Ard, and J. Burdick, "Sensor-based exploration: Incremental construction of the hierarchical generalized Voronoi graph," *The Int. Journal of Robotics Research*, vol. 19, pp. 96-125, Feb. 2000.
- [14] J. Barraquand and J. C. Latombe, "Robot motion planning: A distributed representation approach," *The Int. Journal of Robotics Research*, vol. 10, pp. 628-649, 1991.
- [15] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *Robotics and Automation, in Proc. 1985 IEEE Int. Conf. on*, vol. 2, pp. 500-505, Mar. 1985.
- [16] R. Volpe and P. Khosla, "Artificial potentials with elliptical isopotential contours for obstacle avoidance," *Decision and Control, 26th IEEE Conf. on*, vol. 26, pp. 180-185, Dec. 1987.
- [17] J. Barraquand, L. E. Kavraki, J. C. Latombe, T. Y. Li, R. Motwani, and P. Raghavan, "A random sampling scheme for robot path planning," *The Int. Journal of Robotics Research*, vol. 17, no. 6, pp. 759-774, 1997.
- [18] K.-Y. Im and S.-Y. Oh, "An extended virtual force field based behavioral fusion with neural networks and evolutionary programming for mobile robot navigation," *Evolutionary Computation, proc of the 2000 Congress on*, vol. 2, pp. 1238-1244, 2000.
- [19] L. E. Kavraki, J. C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," in *IEEE Int. Conf. on Methods and Models in Automation and Robotics*, pp. 566-580, Morgan Kaufmann, 1997.
- [20] M. H. Overmars and P. Svestka, "A probabilistic learning approach to motion planning," in *Proc. Workshop on Algorithmic Foundation of Robotics*, pp. 19-38, 1994.
- [21] N. Nilsson, "A mobile automation: An application of artificial intelligence techniques," Tech. Reports 40, AI Center, SRI International, 333 Ravenswood Ave, Menlo Park, CA 94025, Mar. 1969.

- [22] D.-T. Lee, "Proximity and reachability in the plane," Ph.D. thesis, Campaign, IL. USA. 1978.
- [23] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*, second edition, Cambridge, MA, USA: MIT Press, 2001.
- [24] 김정태, 김대진, "가시성 그래프와 가변-셀-분할을 이용한 새로운 경로계획," in *Proc. of Korea Computer Congress (KCC)*, vol. 36, no. 1, pp. 357-361, 2009.
- [25] 김정태, 김대진, "가시성 그래프와 가변-셀-분할을 이용한 경로계획 알고리즘," *한국 자동 제어학술회의 (KACC 2009)*, pp. 283-288, 2009.
- [26] M. de Berg, O. Cheong, M. van Kreveld, and M. H. Overmars, *Computational Geometry: Algorithms and Applications*. Heidelberg: Springer-Verlag, 2008.



김정태

2002년 포항공과대학교 컴퓨터공학과 졸업. 2006년~현재 포항공과대학교대학원 컴퓨터공학과 통합과정 재학중. 관심분야는 로봇공학, 경로 계획, 모션 계획, 기계학습.



김대진

1982년 연세대학교 전자공학과(공학사). 1984년 KAIST 전기 및 전자 공학과(공학석사). 1991년 Syracuse University 전기 및 컴퓨터공학과(공학박사). 1992년~1999년 동아대학교 컴퓨터공학과 부교수. 1999년~현재 포항공과대학교 컴퓨터공학과 교수. 관심분야는 영상 처리, 컴퓨터 비전, 자동 얼굴 및 사람 분석.