



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2009-0100995
(43) 공개일자 2009년09월24일

(51) Int. Cl.

G01C 21/00 (2006.01) G01C 21/34 (2006.01)

(21) 출원번호 10-2008-0026582

(22) 출원일자 2008년03월21일

심사청구일자 2008년03월21일

(71) 출원인

고려대학교 산학협력단

서울 성북구 안암동5가1 고려대학교 내

(72) 발명자

황인준

서울 성북구 돈암동 한진아파트 202동 604호

박진규

서울 동대문구 휘경2동 286-129

(74) 대리인

전중학

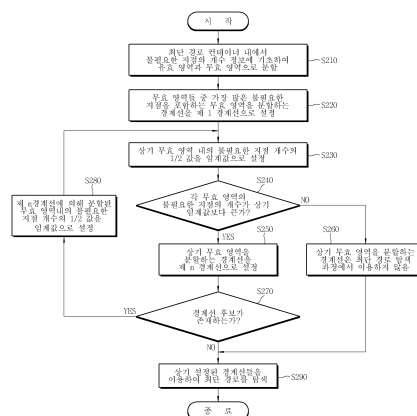
전체 청구항 수 : 총 9 항

(54) 최단 경로 탐색 방법

(57) 요약

본 발명은, 최단 경로로 도달할 수 있는 목적지들의 지점들로 이루어진 도형 내에서 불필요한 지점의 개수를 확인하는 단계와 상기 불필요한 지점의 개수에 기초하여 상기 도형 내 영역을 유효 영역과 무효 영역으로 분할하는 단계 및 상기 유효 영역에 대한 정보를 이용하여 최단 경로를 탐색하는 단계를 포함하되, 상기 유효 영역은 무효 영역보다 상기 불필요한 지점의 개수가 적은 영역을 의미하는 것을 특징으로 하는 최단 경로 탐색 방법을 제공한다.

대표도 - 도2



특허청구의 범위

청구항 1

최단 경로로 도달할 수 있는 목적지들의 지점들로 이루어진 도형 내에서 불필요한 지점의 개수를 확인하는 단계;

상기 불필요한 지점의 개수에 기초하여 상기 도형 내 영역을 유효 영역과 무효 영역으로 분할하는 단계;

상기 유효 영역에 대한 정보를 이용하여 최단 경로를 탐색하는 단계

를 포함하되,

상기 유효 영역은 무효 영역보다 상기 불필요한 지점의 개수가 적은 영역을 의미하는 것을 특징으로 하는 최단 경로 탐색 방법.

청구항 2

제 1항에 있어서,

상기 도형은 최소 경계 직사각형(Minimum Bounding Rectangle)인 것을 특징으로 하는 최단 경로 탐색 방법.

청구항 3

제 2항에 있어서,

상기 최소 경계 직사각형 내에서 최단 경로를 이루는 유효한 지점만으로 컨벡스 헐(convex hull)을 구성하는 단계

를 더 포함하되,

상기 최소 경계 직사각형 내 영역은 상기 컨벡스 헐의 각 변을 연장한 경계선에 의해 분할되는 것을 특징으로 하는 최단 경로 탐색 방법.

청구항 4

제 1항에 있어서,

분할된 무효 영역들 중, 가장 많은 불필요한 지점을 포함하는 무효 영역을 분할한 경계선을 제 1 경계선으로 설정하는 단계

를 더 포함하는 것을 특징으로 최단 경로 탐색 방법.

청구항 5

제 4항에 있어서,

상기 무효 영역 내의 불필요한 지점 개수의 1/2 값을 임계값으로 설정하는 단계와

상기 분할된 무효 영역들 중 불필요한 지점의 개수가 상기 임계값보다 큰 무효 영역의 경우, 상기 무효 영역을 분할한 경계선을 제 n 경계선으로 설정하는 단계

를 더 포함하되,

상기 설정된 n개의 경계선들을 이용하여 최단 경로를 탐색하는 것을 특징으로 하는 최단 경로 탐색 방법.

청구항 6

제 1항에 있어서,

상기 유효 영역과 상기 무효 영역을 구별하는 플래그 정보를 설정하는 단계를 더 포함하되,

상기 플래그 정보에 기초하여 상기 도형 내 영역을 유효 영역과 무효 영역으로 분할하는 것을 특징으로 하는 최단 경로 탐색 방법.

청구항 7

제 1항에 있어서,

상기 유효 영역은 불필요한 지점을 포함할 수 있는 것을 특징으로 하는 최단 경로 탐색 방법.

청구항 8

제 1항에 있어서,

상기 무효 영역은 유효한 지점을 포함할 수 없는 것을 특징으로 하는 최단 경로 탐색 방법.

청구항 9

경로 탐색의 시작 지점의 이웃에 위치하는 지점들의 최소 경계 직사각형(Minimum Bounding Rectangle)을 설정하는 단계;

상기 최소 경계 직사각형 내에 목적지가 포함되어 있다면, 상기 이웃 지점의 최소 경계 직사각형을 유효 영역과 무효 영역으로 분할하는 단계;

상기 목적지가 상기 유효 영역 내에 포함되어 있는 경우, 상기 이웃 지점을 큐(queue)에 저장하는 단계를 포함하되,

상기 유효 영역은 무효 영역보다 불필요한 지점의 개수가 적은 영역을 의미하는 것을 특징으로 하는 최단 경로 탐색 방법.

명세서

발명의 상세한 설명

기술 분야

<1> 본 발명은 최단 경로를 탐색하는 방법에 관한 것이다.

배경 기술

<2> 최단 경로 컨테이너(Shortest path containers, SPC) 기반의 경로 탐색 알고리즘에서 최소 경계 직사각형(Minimum Bounding Rectangle, MBR)을 컨테이너(container)로 사용하는 경우, 보다 빠른 검색 속도를 얻을 수 있다. 하지만, 최소 경계 직사각형 내부에 불필요한 지점을 포함하게 될 경우, 경로 탐색 과정에서 불필요한 이웃 지점을 방문하게 되어 검색 시간이 늘어나게 된다.

발명의 내용

해결 하고자하는 과제

<3> 본 발명은, 두 지점 간의 최단 경로를 탐색하는 방법을 제공한다.

<4> 또한, 본 발명은, 최소 경계 직사각형(Minimum Bounding Rectangle) 내부의 불필요한 방문 지점의 수를 줄이고자 한다.

<5> 또한, 본 발명은, 경로 탐색 알고리즘에서 보다 빠른 검색 속도를 얻고자 한다.

과제 해결수단

<6> 본 발명은, 최단 경로로 도달할 수 있는 목적지들의 지점들로 이루어진 도형 내에서 불필요한 지점의 개수를 확인하는 단계와 상기 불필요한 지점의 개수에 기초하여 상기 도형 내 영역을 유효 영역과 무효 영역으로 분할하는 단계 및 상기 유효 영역에 대한 정보를 이용하여 최단 경로를 탐색하는 단계를 포함하되, 상기 유효 영역은 무효 영역보다 상기 불필요한 지점의 개수가 적은 영역을 의미하는 것을 특징으로 하는 최단 경로 탐색 방법을 제공한다.

<7> 또한, 본 발명은, 최단 경로로 도달할 수 있는 목적지들의 지점들로 이루어진 도형은 최소 경계 직사각형

(Minimum Bounding Rectangle)인 것을 특징으로 한다.

- <8> 또한, 본 발명은, 최소 경계 직사각형 내에서 최단 경로를 이루는 유효한 지점만으로 컨벡스 헐(convex hull)을 구성하는 단계를 더 포함하되, 상기 최소 경계 직사각형 내 영역은 상기 컨벡스 헐의 각 변을 연장한 경계선에 의해 분할되는 것을 특징으로 한다.
- <9> 또한, 본 발명은, 분할된 무효 영역들 중, 가장 많은 불필요한 지점을 포함하는 무효 영역을 분할하는 경계선을 제 1 경계선으로 설정하는 단계를 더 포함하는 것을 특징으로 한다.
- <10> 또한, 본 발명은, 상기 무효 영역 내의 불필요한 지점 개수의 1/2 값을 임계값으로 설정하는 단계와 상기 분할된 무효 영역들 중 불필요한 지점의 개수가 상기 임계값보다 큰 무효 영역의 경우, 상기 무효 영역을 분할하는 경계선을 제 n 경계선으로 설정하는 단계를 더 포함하되, 상기 설정된 n개의 경계선들을 이용하여 최단 경로를 탐색하는 것을 특징으로 한다.
- <11> 또한, 본 발명은, 상기 유효 영역과 상기 무효 영역을 구별하는 플래그 정보를 설정하는 단계를 더 포함하되, 상기 플래그 정보에 기초하여 상기 도형 내 영역을 유효 영역과 무효 영역으로 분할하는 것을 특징으로 한다.
- <12> 또한, 본 발명은, 상기 유효 영역은 불필요한 지점을 포함할 수 있는 것을 특징으로 한다.
- <13> 또한, 본 발명은, 상기 무효 영역은 유효한 지점을 포함할 수 없는 것을 특징으로 한다.
- <14> 또한, 본 발명은, 경로 탐색의 시작 지점의 이웃에 위치하는 지점들의 최소 경계 직사각형(Minimum Bounding Rectangle)을 설정하는 단계와 상기 최소 경계 직사각형 내에 목적지가 포함되어 있다면, 상기 이웃 지점의 최소 경계 직사각형을 유효 영역과 무효 영역으로 분할하는 단계 및 상기 목적지가 상기 유효 영역 내에 포함되어 있는 경우, 상기 이웃 지점을 큐(queue)에 저장하는 단계를 포함하되, 상기 유효 영역은 무효 영역보다 불필요한 지점의 개수가 적은 영역을 의미하는 것을 특징으로 하는 최단 경로 탐색 방법을 제공한다.

효 과

- <15> 본 발명에서는 최소 경계 직사각형(Minimum Bounding Rectangle) 내부의 불필요한 방문 지점의 수를 줄임으로써 보다 빠른 속도로 경로를 탐색할 수 있다. 또한, 경로 탐색 과정에서 경계선(Border Line)을 이용함으로써 경로 검색 시간을 줄일 수 있다.

발명의 실시를 위한 구체적인 내용

- <16> 본 발명이 적용되는 최단 경로를 탐색하는 방법은 네비게이션, 통신 네트워크, 자율 이동 로봇, 게임 소프트웨어, 인공 지능 등 다양한 분야에서 이용될 수 있다. 이 때, 상기 최단 경로를 계산하는 방법으로 다양한 알고리즘을 적용할 수 있다. 또한, 본 명세서에서 지점이란 통상적인 장소의 의미뿐만 아니라 알고리즘에 쓰이는 노드의 의미를 포함할 수 있다.
- <17> 예를 들어, 다익스트라(Dijkstra) 알고리즘을 들 수 있다. 상기 다익스트라 알고리즘은 특정한 두 지점 간의 최단 경로를 계산하는 것이 아니다. 시작점에서부터, 이웃한 모든 주변 지점 간의 최단 경로를 계산할 수 있다.
- <18> 다른 예로, 직접 목표(Goal-directed) 탐색 방법은 모든 이웃 지점을 방문하는 것이 아니라 각 이웃 지점으로부터 목적지까지의 거리를 예측하여 상기 목적지에 가까운 지점을 방문하도록 할 수 있다.
- <19> 다른 예로, 지도상의 모든 두 지점 간의 최단 경로를 구한 뒤, 모든 최단 경로를 각 에지 기준으로 분류하여 최단 경로 컨테이너(Shortest-path containers)에 저장할 수 있다. 그리고, 경로 계산 과정에서 상기 저장된 최단 경로를 참조하여 어느 이웃 지점이 목적지와 최단 거리인지를 확인할 수 있다.
- <20> 다른 예로, 프로이드 알고리즘의 경우, 출발 지점에서 목적 지점까지 가능한 모든 링크를 검사한다. 그리고 상기 링크들을 비교하여 최단 경로를 탐색할 수 있다.
- <21> 다른 예로, A*알고리즘은 현재 지점에서 목적 지점까지의 잔여 경로 비용의 최소 추정치를 이용하여 최단 경로를 추정할 수 있다. 즉, 출발 지점에서 현재 지점까지의 실거리와 목적 지점까지의 추정 거리를 이용하여 최단 경로를 탐색함으로써 탐색 시간을 단축시킬 수 있다.
- <22> 이하, 상기 다양한 알고리즘들이 적용될 수 있는 최단 경로 탐색 방법에 대한 구체적인 실시예들을 살펴보기로 한다.

- <23> 도 1은 본 발명이 적용되는 실시예로서, 최단 경로 탐색 방법을 설명하기 위한 개략적인 흐름도를 나타낸다.
- <24> 먼저, 최단 경로를 탐색하기에 앞서 지도 정보를 획득할 수 있다(S110).
- <25> 모든 경로를 미리 계산하여 데이터베이스에 저장하는 경우, 시작 지점과 목적지를 질의로 하여 데이터베이스에서 경로를 찾는 방식으로 최단 경로를 구할 수 있다(S120). 또한, 상기 모든 경로 정보를 각 에지를 통해 최단 경로로 도달할 수 있는 모든 지점의 집합 형태로 저장하게 된다면 저장 공간도 최소화할 수 있다. 이 때, 상기 최단 경로로 도달할 수 있는 지점의 집합은 최단 경로 컨테이너(Shortest-path containers)에 저장될 수 있다. 예를 들어, 상기 최단 경로 컨테이너는 최소 경계 직사각형으로 구성될 수 있다(S130).
- <26> 상기 최소 경계 직사각형 내에서 불필요한 지점의 개수에 기초하여 유효 영역과 무효 영역으로 분할할 수 있다. 이때, 상기 유효 영역과 상기 무효 영역으로 분할하는 직선을 경계선이라 부르기로 한다. 이처럼, 상기 최소 경계 직사각형 내에서 유효 영역과 무효 영역을 분할하는 경계선을 생성할 수 있다(S140). 그리고, 상기 경계선들을 경로 탐색 데이터 베이스에 저장할 수 있다(S150). 상기 저장된 정보들을 이용하여 최단 경로를 탐색할 수 있게 된다.
- <27> 이하, 상기 경계선을 생성하는 구체적인 방법들을 살펴보도록 한다.
- <28> 도 2는 본 발명이 적용되는 실시예로서, 경계선을 이용하여 최단 경로를 탐색하는 방법을 설명하기 위해 나타낸 흐름도이다.
- <29> 경로 탐색 과정에서 현재 지점의 모든 이웃 지점에 대한 컨테이너에 목적지가 포함되어 있는지 여부를 확인함으로써, 어느 이웃 지점을 통하는 것이 최단 경로로 목적지까지 도달할 수 있는지를 판단할 수 있다. 뿐만 아니라, 불필요한 이웃 지점 방문을 최소화하여 검색 시간도 단축시킬 수 있다.
- <30> 여기서, 상기 최단 경로 컨테이너(Shortest-path containers)는 여러 형태의 도형으로 구성될 수 있다. 예를 들어, 상기 최단 경로 컨테이너(Shortest-path containers)의 도형으로 모든 지점을 포함하는 최소 경계 직사각형(Minimum Bounding Rectangle)과 컨벡스 헐(Convex Hull)을 들 수 있다. 상기 최소 경계 직사각형(Minimum Bounding Rectangle)의 경우 컨테이너(container) 내부에 불필요한 지점을 포함할 수 있다. 예를 들어, 특정 에지를 통해 최단 거리로 도달하지는 못하지만 상기 최소 경계 직사각형(Minimum Bounding Rectangle)을 구성하는 과정에서 상기 최소 경계 직사각형(Minimum Bounding Rectangle)에 포함되는 지점을 포함할 수 있다. 이것은 이후 경로 찾기 과정에서 불필요한 이웃 지점으로 방문을 허용하여 탐색 시간이 늘어나는 문제가 될 수 있다.
- <31> 상기 컨벡스 헐(convex hull)의 경우, 상기 최소 경계 직사각형(Minimum Bounding Rectangle)에서의 불필요한 지점의 수를 최소화할 수 있다. 그러나, 상기 컨벡스 헐(convex hull)의 경우 경로 탐색 과정에서 목적지가 각 컨벡스 헐(convex hull) 내부에 존재하는지 여부를 확인하는 비용이 클 수 있다. 반면, 상기 최소 경계 직사각형의 경우 사각형 내부에 목적지가 포함되어 있는지 여부를 확인하는 루틴은 매우 간단하므로 불필요한 이웃 지점의 방문이 있더라도 컨벡스 헐(convex hull) 보다 빠른 검색 속도를 나타낼 수 있다.
- <32> 또한, 상기 컨벡스 헐(convex hull)의 경우, 불필요한 방문 지점의 수를 최소화할 수 있는 컨테이너(container)이다. 그러나, 컨벡스 헐(convex hull) 내부에 한점이 포함되는지 여부를 확인하는 루틴이 복잡할 수 있다. 따라서, 본 발명에서는 상기 최소 경계 직사각형(Minimum Bounding Rectangle)과 상기 컨벡스 헐(convex hull)을 모두 이용하는 경계선 기반의 최적 경로 탐색 방안(Border Line-Based Routing Scheme)을 제안한다.
- <33> 먼저, 최단 경로 컨테이너 내에서 불필요한 지점의 개수 정보에 기초하여 유효 영역과 무효 영역으로 분할할 수 있다(S210). 이때, 최소 경계 직사각형(Minimum Bounding Rectangle) 내부에서 불필요한 지점이 밀집된 지역을 무효 영역(Invalid Region)이라 하고, 그 이외의 지역을 유효 영역(Valid Region)이라고 정의할 수 있다. 상기 두 영역의 특징을 설명하면, 상기 유효 영역(Valid Region)에는 불필요한 지점이 포함될 수 있으나 상기 무효 영역(Invalid Region)에는 유효한 지점이 포함될 수 없다. 그리고, 상기 두 영역을 최대 k개의 직선을 사용하여 나눌 때 상기 직선을 경계선(Border Line)이라고 정의한다. 상기 최소 경계 직사각형(Minimum Bounding Rectangle)을 상기 유효 영역(Valid Region)과 여러 개의 무효 영역(Invalid Region)으로 분할할 때 컨벡스 헐(convex hull)을 사용할 수 있다. 상기 컨벡스 헐(convex hull)을 이용하는 경우, 상기 무효 영역(Invalid Region)과 상기 유효 영역(Valid Region) 특성을 만족할 수 있다.
- <34> 본 발명이 적용되는 실시예로서, 무효 영역(Invalid Region)과 유효 영역(Valid Region)을 구분하는 경계선(Border Line)을 생성하는 방법을 설명하도록 한다.
- <35> 먼저, 컨벡스 헐(convex hull)의 불필요한 각 변을 최소 경계 직사각형(Minimum Bounding Rectangle)까지 확장

한 직선을 경계선(Border Line) 후보로 설정할 수 있다. 각 직선에 의해 구분된 무효 영역(Invalid Region)에 대해 각 무효 영역(Invalid Region)에 포함되는 불필요한 지점의 수를 계산하고, 상기 불필요한 지점의 개수를 기준으로 각 직선을 정렬할 수 있다. 그리고, 불필요한 지점을 가장 많이 제외하는 직선을 제 1 경계선(Border Line)으로 설정할 수 있다. 이후 각 직선에 대해 제외되는 불필요한 지점의 수를 누적하면서 증가분이 상기 첫 번째 경계선(Border Line)으로 제외된 지점 수의 1/2 보다 큰 경우 추가적인 경계선(Border Line)으로 설정할 수 있다.

<36> 경로 탐색 과정에서는 현재 위치의 모든 이웃 지점에 대하여 각 에지의 최소 경계 직사각형(Minimum Bounding Rectangle)에 목적지가 포함되어 있는지 여부를 확인한다. 만약 상기 최소 경계 직사각형(Minimum Bounding Rectangle) 내부에 목적지가 있는 경우 상기 최소 경계 직사각형(Minimum Bounding Rectangle)내의 경계선들(Border Lines)을 확인하여 목적지가 유효 영역(Valid Region)에 포함되는지 여부를 확인할 수 있다.

<37> 한편, 경로 탐색에 사용하기 위한 경계선(Border Line)을 생성하기 위하여, 지도 정보로부터 모든 시작 지점에서부터 목적지 사이의 최단 경로를 계산할 수 있다. 상기 계산된 최단 경로 정보에 기초하여, 각 에지를 통해 최단 경로로 도달할 수 있는 목적지들의 집합을 최소 경계 직사각형(Minimum Bounding Rectangle) 형태로 계산할 수 있다. 상기 각 최소 경계 직사각형(Minimum Bounding Rectangle)에서 불필요한 노드를 제거하기 위하여, 유효한 노드들만으로 컨벡스 쉘(convex hull)을 구성할 수 있다. 그리고, 상기 컨벡스 쉘(convex hull)의 각 변을 경계선(Border Line)의 후보로 설정할 수 있다. 상기 후보로 설정된 경계선(Border Line)에 의해 제거되는 불필요한 노드의 개수를 확인할 수 있다. 그리고, 상기 불필요한 노드의 개수에 기초하여, 상기 후보로 설정된 경계선들을 정렬할 수 있다. 이때, 가장 많이 불필요한 노드를 제외하는 경계선 후보를 제 1 경계선(first border line)으로 설정할 수 있다(S220).

<38> 그리고, 상기 경계선(Border Line)으로 제외되는 불필요한 노드 수의 1/2 값을 임계값(threshold)으로 설정할 수 있다(S230). 이후, 각 경계선 후보에 대하여 제외되는 노드의 개수를 확인할 수 있다. 이때, 이미 제외되었던 노드는 제외할 수 있다. 상기 각 경계선 후보에 대하여 제외되는 노드의 개수와 상기 임계값을 비교할 수 있다(S240). 현재 경계선 후보에 대하여 제외되는 노드의 개수가 상기 임계값보다 클 경우, 상기 현재 경계선 후보는 제 n 경계선(Border Line)으로 추가적으로 설정할 수 있다(S250). 그리고, 경계선 후보가 남아있는 경우(S270), 상기 제 n경계선에 의해 분할된 무효 영역 내의 불필요한 지점 개수의 1/2 값을 다시 새로운 임계값으로 설정할 수 있다(S280). 이후, 상기 S240단계와 S250단계를 반복해서 수행할 수 있다. 그러나, 상기 현재 경계선 후보에 대하여 제외되는 노드의 개수가 상기 임계값보다 크지 않을 경우에는, 상기 현재 경계선 후보는 최단 경로 탐색 과정에서 이용되지 않을 수 있다(S260).

<39> 한편, 각 경계선(Border Line)이 추가적으로 설정될 때마다, 직선으로 나누어지는 두 영역에 대해서 어느 쪽이 불필요한 영역인지를 판단하는 플래그를 설정할 수 있다. 이렇게 생성된 최소 경계 직사각형(Minimum Bounding Rectangle) 정보와 경계선(Border Line) 정보를 경로 탐색 데이터 베이스에 추가할 수 있다.

<40> 상기 S270단계에서 더이상 경계선 후보가 존재하지 않는 경우에는, 상기 경로 탐색 데이터 베이스에 저장된 정보들을 이용하여 최단 경로를 탐색할 수 있다(S290).

<41> 도 3은 본 발명이 적용되는 실시예로서, 경로 탐색 과정에서 경계선을 확인하는 과정을 설명하기 위한 흐름도를 나타낸다.

<42> 일반적으로 선형리스트에서 데이터의 삽입 및 제거 형식이 먼저 삽입된 것이 먼저 제거되는 이른바 선입선출에 따르는 데이터 구조를 큐라 한다. 이것과는 달리, 데이터마다 각각 우선순위가 정의되어 있는 경우에 데이터가 삽입된 순위로는 제거되지 않고, 순위가 높은 것(값이 작은 것)부터 먼저 제거되는 데이터구조를 우선순위 큐(Priority Queue)라 한다. 상기 우선순위 큐에서 우선순위인 종류가 적은 경우에는 우선순위마다 큐를 마련한 데이터구조를 이용하는 경우가 많다. 우선순위뿐만 아니라 도착 시간도 함께 내용(contents)으로서 데이터에 곁들여 기록되어 있는 경우에는 내용이 최소인 요소를 살펴서 데이터를 제거하면 된다.

<43> 경로 탐색 과정에서 먼저 시작 지점을 우선순위 큐(Priority Queue)에 추가할 수 있다(S300). 상기 큐(Queue)의 맨 위에 저장된 지점에 대하여, 상기 지점의 이웃에 위치하는 지점들의 목록을 확인할 수 있다(S310). 그리고, 상기 각 이웃 지점에 대하여 최소 경계 직사각형(Minimum Bounding Rectangle)을 구성할 수 있다(S320). 목적지가 상기 최소 경계 직사각형(Minimum Bounding Rectangle) 내부에 있지 않다면(S330), 상기 이웃 지점은 경로 탐색 과정에서 더 이상 고려하지 않을 수 있다(S360).

<44> 반면, 상기 목적지가 상기 최소 경계 직사각형(Minimum Bounding Rectangle) 내부에 있다면(S330), 상기 목적

지가 상기 현재 이웃 지점의 경계선에 의해 분할된 유효 영역에 있는지 여부를 확인할 수 있다(S340). 상기 목적지가 상기 이웃 지점 경계선의 유효 영역에 있을 때 상기 이웃 노드를 큐(Queue)에 추가할 수 있다(S350). 그러나, 상기 목적지가 상기 이웃 지점 경계선의 유효 영역에 존재하지 않는 경우, 상기 이웃 지점은 경로 탐색 과정에서 더 이상 고려하지 않을 수 있다(S360). 그리고, 상기 과정을 상기 큐(Queue)가 비워질 때까지 반복하여 목적지까지의 최단 거리를 구할 수 있다.

<45> 이상, 전술한 본 발명의 바람직한 실시예는, 예시의 목적을 위해 개시된 것으로, 당업자라면 이하 첨부된 특허 청구범위에 개시된 본 발명의 기술적 사상과 그 기술적 범위 내에서, 다양한 다른 실시예들을 개량, 변경, 대체 또는 부가 등이 가능할 것이다.

도면의 간단한 설명

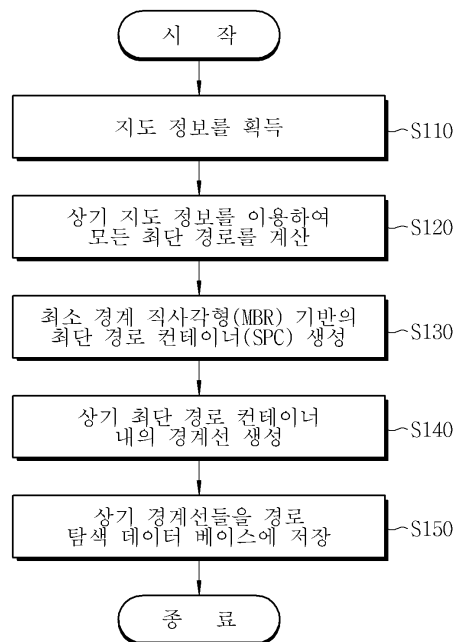
<46> 도 1은 본 발명이 적용되는 실시예로서, 최단 경로 탐색 방법을 설명하기 위한 개략적인 흐름도를 나타낸다.

<47> 도 2는 본 발명이 적용되는 실시예로서, 경계선을 이용하여 최단 경로를 탐색하는 방법을 설명하기 위해 나타낸 흐름도이다.

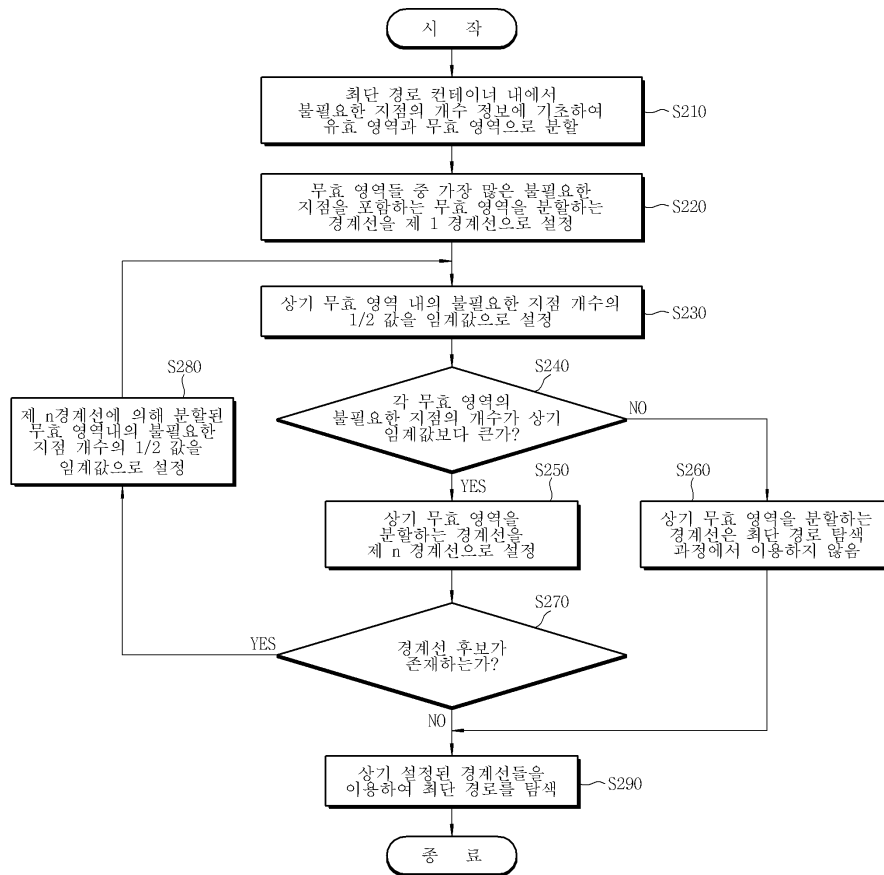
<48> 도 3은 본 발명이 적용되는 실시예로서, 경로 탐색 과정에서 경계선을 확인하는 과정을 설명하기 위한 흐름도를 나타낸다.

도면

도면1



도면2



도면3

