



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2013-0094265
(43) 공개일자 2013년08월23일

(51) 국제특허분류(Int. Cl.)
G06F 17/30 (2006.01) G06F 17/40 (2006.01)
(21) 출원번호 10-2013-0059785(분할)
(22) 출원일자 2013년05월27일
심사청구일자 2013년05월27일
(62) 원출원 특허 10-2012-0015153
원출원일자 2012년02월15일
심사청구일자 2012년02월15일

(71) 출원인
인하대학교 산학협력단
인천광역시 남구 인하로 100, 인하대학교 (용현동)
(72) 발명자
신승선
인천광역시 남구 용현4동 하이테크 1008
배해영
인천광역시 연수구 송도동 더샵퍼스트월드 1동 201호
(뒷면에 계속)
(74) 대리인
이은철

전체 청구항 수 : 총 3 항

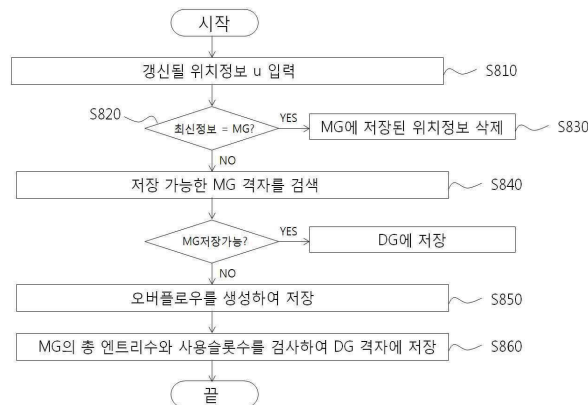
(54) 발명의 명칭 도로 네트워크 환경에서의 색인 갱신방법

(57) 요약

본 발명은 도로 네트워크 환경에서의 색인방법 및 갱신방법에 관한 것으로, 보다 상세하게는 도로망을 격자로 분할하여 도로 연결정보를 색인구조에 포함하고, 고정 객체를 별도의 색인으로 유지하여 도로네트워크 공간에서 이동 객체의 현재 위치를 효율적으로 관리하기 위한 도로 네트워크 환경에서의 색인 갱신방법에 관한 것이다.

이러한 목적을 달성하기 위한 본 발명은, (a) 입력부가 갱신될 새로운 위치정보를 입력받는 단계; (b) 검색부가 새로운 위치정보가 기존 메모리 격자(MG)에 저장되어 있는지 검색하는 단계; 및 (c) 상기 (b) 단계의 검색결과, 상기 새로운 위치정보가 저장되어 있는 경우, 색인 갱신부가 상기 기존 메모리 격자(MG)에 저장되어 있는 위치정보를 삭제하고, 상기 새로운 위치정보를 저장하여 색인을 갱신하는 단계; 를 포함한다.

대표도 - 도8



(72) 발명자

이연

인천광역시 남구 용현동 인하대학교

백성하

서울특별시 금천구 시흥동 852-37 2층

김태규

경기도 부천시 원미구 상동 목련마을 상록아파트
2806동 204호

이 발명을 지원한 국가연구개발사업

과제고유번호 07 국토정보 C05

부처명 국토해양부

연구사업명 첨단도시기술개발사업

연구과제명 지능형 GIS 연구센터 활성화 추진연구

주관기관 인하대학교 산학협력단

연구기간 2011.04.27 ~ 2012.04.26

특허청구의 범위

청구항 1

격자색인정보와 관심객체색인정보가 통합된 색인정보를 포함하는 도로 네트워크 환경에서의 색인 갱신방법에 있어서,

- (a) 입력부(100)가 갱신될 새로운 위치정보를 입력받는 단계;
- (b) 검색부(200)가 새로운 위치정보가 기존 메모리 격자(MG)에 저장되어 있는지 검색하는 단계; 및
- (c) 상기 (b) 단계의 검색결과, 상기 새로운 위치정보가 저장되어 있는 경우, 색인 갱신부(300)가 상기 기존 메모리 격자(MG)에 저장되어 있는 위치정보를 삭제하고, 상기 새로운 위치정보를 저장하여 색인을 갱신하는 단계; 를 포함하는 도로 네트워크 환경에서의 색인 갱신방법.

청구항 2

제 1 항에 있어서,

- (d) 상기 (b) 단계의 검색결과, 저장되어 있지 않을 경우, 메모리 격자 검색부(400)가 상기 새로운 위치정보가 저장될 수 있는 메모리 격자를 검색하는 단계;
- (e) 상기 (d) 단계의 검색결과, 디스크 격자 모든 페이지에 저장할 공간이 없는 경우, 오버 플로우 생성부(500)가 오버 플로우를 생성하여 저장하는 단계; 및
- (f) 위치정보 저장부(600)가 메모리 격자의 총 엔트리수와 사용슬롯수를 검사하여 상기 (e) 단계를 통해 생성된 오버 플로우에 새로운 위치정보를 저장하는 단계; 를 포함하는 것을 특징으로 하는 도로 네트워크 환경에서의 색인 갱신방법.

청구항 3

제 2 항에 있어서,

상기 (f) 단계는,

- (f-1) 상기 위치정보 저장부(600)가 메모리 격자에 저장되어 있는 디스크격자 페이지 정보 중 삭제대상 메모에 저장되어 있는 위치정보를 가져온 디스크격자를 삭제하는 단계;
- (f-2) 상기 위치정보 저장부(600)가 상기 메모리격자에 저장되어 있는 최신 위치정보를 읽어온 디스크격자에 저장하는 단계;
- (f-3) 상기 위치정보 저장부(600)가 디스크에 쓰기를 실행하는 단계; 를 포함하는 도로네트워크 환경에서의 색인 갱신방법.

명세서

기술분야

[0001] 본 발명은 도로 네트워크 환경에서의 색인 갱신방법에 관한 것으로, 보다 상세하게는 도로망을 격자로 분할하여 도로 연결정보를 색인구조에 포함하고, 고정 객체를 별도의 색인으로 유지하여 도로네트워크 공간에서 이동 객체의 현재 위치를 효율적으로 관리하기 위한 도로 네트워크 환경에서의 색인 갱신방법에 관한 것이다.

배경기술

[0002] 이동 통신 기술이 발달함에 따라 GPS 기술의 무료 개방 및 무선 인터넷 망의 개방과 세계 각국의 정책적 지원에 힘입어 위치 기반 서비스(Location-based service)가 이동통신 분야에서 새로운 기반 기술로 각광받고 있다.

위치 기반 서비스는 GPS와 같은 기술을 사용하여 이용자의 위치를 파악하고 이와 관련된 다양한 응용 서비스를 지원한다. 위치 기반 서비스에서 적용 가능한 서비스 분야에는 자산 추적, 차량 추적, 교통량 감시 시스템, 주변정보 조회, 내비게이션과 같은 생활 편의를 위한 서비스들이 있다. 또한, 위치 기반 서비스는 하천 모니터링, 화재 감시 모니터링과 같은 u-방재 서비스의 기반 기술로도 사용되고 있다. 이와 같은 서비스를 지원하기 위해서는 시간의 흐름에 따라 위치가 변화하는 시공간 데이터(spatio-temporal data)의 특성을 지니는 이동 객체의 위치 정보를 효과적으로 저장 및 검색하기 위한 기술이 필수적이다.

[0003] 이동 객체는 주기적으로 자신의 위치 정보를 서버로 보내게 되고 서버는 이동 객체의 갱신되는 위치 정보를 저장하게 된다. 이때 저장되는 데이터는 각 이동 객체에 대해 전송되는 위치 정보를 연속적으로 저장하거나(과거 궤적정보), 현재 위치 정보만을 저장한다(현재위치정보). 또한 이동 객체의 위치 정보와 함께 속도, 방향 등과 같은 부가적인 정보를 함께 저장(미래위치예측)하거나 각 응용에 맞게 위치 정보를 혼합하여 저장한다. 이렇게 저장된 과거, 현재, 미래 위치 정보는 범위 질의(range query), k-최근접 질의(k-nearest neighbor query), 궤적 질의(trjectory query)와 같은 질의를 효율적으로 처리할 수 있는 색인 구조에 대한 많은 연구가 진행되고 있다. 연구되고 있는 색인은 크게 색인에 적용되는 공간 모델링을 사용하여 하늘과 같이 이동 객체의 움직임에 제약이 없는 공간인 유클리드 공간과 움직임의 제약이 강한 도로 네트워크 공간에서 사용되는 색인으로 나눌 수 있다. 지금까지 연구되었던 색인의 대부분은 대량의 이동 객체에 대한 갱신 시의 비용 문제를 여전히 가지고 있다. 유클리드 공간 색인의 경우에는 갱신 문제를 갱신 지연 방법 등을 적용하여 해결하였지만 대량의 이동 객체에 대한 갱신 요청이 극히 짧은 주기로 발생한 경우 성능이 저하되는 문제점이 존재한다. 그리고 움직임의 제약이 없는 유클리드 공간에서 연구가 진행되었기 때문에 건물, 도로와 와 같이 이동 객체의 움직임에 제약이 강한 실세계에 적용하기 어렵고 이동 객체의 위치 정보를 갱신하는 경우, 이미 저장되어 있는 이동 객체를 빠르게 찾기 위해 해싱(hashing)과 같은 보조 색인을 사용하게 되는데, 이때 사용되는 보조 색인의 유지비용이 크다는 문제점이 있다. 또한, 유클리드 공간에서의 색인에서 관리 되는 이동 객체는 인접한 도로로 이동해야 하는 특성을 갱신 연산이나 질의 처리 시에 활용하지 못하여 비효율적이다. 도로 네트워크 공간 색인의 경우에도 빈번한 갱신 문제, 보조 색인의 유지비용 문제나 도로의 연결정보를 활용하지 못한다는 문제점이 존재한다.

[0004] 이러한 문제점을 해결하기 위해서 한국등록특허 제10-0709274호 (발명의 명칭: 도로 네트워크 기반의 이동체의 통합 색인구조)는 도로 네트워크 기반의 이동체의 현재 위치 색인 및 과거 위치 색인의 공통된 부분인 망 구조를 저장 및 관리하는 부분을 공유하고, 현재 위치에 대한 색인, 과거 위치에 대한 색인 및 현재 위치에 의 갱신으로 발생하는 가까운 과거 위치에 대한 생성 및 처리 기능을 통합한 색인으로, 이동체의 현재 위치 뿐만아니라 과거 위치까지 함께 지원하여 색인의 공간적인 비용 및 갱신비용을 감소시키는 발명을 제안한 바 있다. 다만, 이러한 발명 역시 이동객체의 빈번한 갱신문제와 보조 색인의 유지비용 문제를 해결하지 못하는 단점이 있고, 검색 성능을 향상하지 못했다는 문제가 있었다.

발명의 내용

해결하려는 과제

[0005] 본 발명은 상기와 같은 문제점을 감안하여 안출된 것으로, 도로 네트워크의 특성을 반영하여 이동객체의 이동방향을 쉽게 예측하고, 도로를 분할하여 도로 세그먼트 단위로 이동 객체들을 관리할 수 있기 때문에 갱신 비용을 줄이고 검색 성능을 향상시킬 수 있는 도로 네트워크 환경에서의 색인 갱신방법을 제공함에 있다.

과제의 해결 수단

[0006] 이러한 기술적 과제를 달성하기 위한 본 발명은 도로 네트워크 환경에서의 색인 갱신방법에 관한 것으로서, (a) 입력부가 갱신될 새로운 위치정보를 입력받는 단계; (b) 검색부가 새로운 위치정보가 기존 메모리 격자(MG)에 저장되어 있는지 검색하는 단계; 및 (c) 상기 (b) 단계의 검색결과, 상기 새로운 위치정보가 저장되어 있는 경우, 색인 갱신부가 상기 기존 메모리 격자(MG)에 저장되어 있는 위치정보를 삭제하고, 상기 새로운 위치정보를 저장하여 색인을 갱신하는 단계; 를 포함한다.

발명의 효과

[0007] 상기와 같은 본 발명에 따르면, 도로 네트워크의 특성을 반영하여 이동객체의 이동방향을 쉽게 예측하고, 도로를 분할하여 도로 세그먼트 단위로 이동 객체들을 관리할 수 있기 때문에 갱신 비용을 줄이고 검색 성능을 향상시킬 수 있다.

도면의 간단한 설명

- [0008] 도 1 은 본 발명의 일실시예에 따른 도로 네트워크 환경에서의 색인방법에 대한 흐름도.
 도 2 는 본 발명의 일실시예에 따른 LUGrid의 전체적인 구조도.
 도 3 은 본 발명의 일실시예에 도로망 분할과정에 대한 도식도.
 도 4 는 본 발명의 일실시예에 따른 도로 네트워크 환경에서의 도로 공간 분할 화면에 대한 도식도.
 도 5a 는 본 발명의 일실시예에 따른 IP를 포함하는 격자의 격자 디렉토리의 일예시도.
 도 5b 는 본 발명의 일실시예에 따른 IP를 포함하지 않는 격자의 격자 디렉토리의 일예시도.
 도 5c 는 본 발명의 일실시예에 따른 삭제대상메모(MDM) 격자의 격자 디렉토리의 일예시도.
 도 6 은 본 발명의 일실시예에 따른 R*-tree를 이용한 POI 색인방법에 대한 도식도.
 도 7 은 본 발명의 일실시예에 따른 색인 통합 방법에 대한 도식도.
 도 8 은 본 발명의 일실시예에 따른 도로 네트워크 환경에서의 색인 갱신방법에 대한 흐름도.
 도 9 는 본 발명의 일실시예에 따른 도로 네트워크 환경에서의 색인방법에 대한 범위 질의 처리방법에 대한 흐름도.
 도 10 은 본 발명의 일실시예에 따른 도로 네트워크 환경에서의 색인방법에 대한 주변 정보 검색 방법의 흐름도.
 도 11 은 본 발명의 일실시예에 따른 도로 네트워크 환경에서의 색인 시스템에 관한 구성도.
 도 12 는 본 발명의 일실시예에 따른 도로 네트워크 환경에서의 색인 갱신 시스템에 관한 구성도.

발명을 실시하기 위한 구체적인 내용

- [0009] 본 발명의 구체적 특징 및 이점들은 첨부도면에 의거한 다음의 상세한 설명으로 더욱 명백해질 것이다. 이에 앞서 본 발명에 관련된 공지 기능 및 그 구성에 대한 구체적인 설명이 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우에는, 그 구체적인 설명을 생략하였음에 유의해야 할 것이다.
- [0010] 이하, 첨부된 도면을 참조하여 본 발명을 상세하게 설명한다.
- [0011] 도 1 은 본 발명의 일실시예에 따른 도로 네트워크 환경에서의 색인방법에 대한 흐름도이며, 도 11 은 본 발명의 일실시예에 따른 도로 네트워크 환경에서의 색인 시스템에 관한 전체 구성도로서, 본 발명의 일실시예에 따른 도로 네트워크 환경에서의 색인 시스템(S)은 분할부(100), 관심객체색인정보 생성부(200) 및 통합부(300)를 포함하여 이루어진다.
- [0012] 분할부(100)는 도로망의 교차점을 기준으로 격자를 나눔으로써 공간을 분할하여, 각각의 공간마다 격자특성을 파악하고, 색인을 생성하여 격자색인정보를 저장한다.
- [0013] 도로네트워크는 도로망이 라인으로 표현되고 도로망이 교차하는 지점에 교차점이 생성되는 특징을 이용하여, 도로망을 기준으로 공간을 분할한다.
- [0014] 구체적으로, 분할부(100)는 도로망의 교차점을 기준으로 공간을 분할한다(S110). 이때, 분할부(100)는 실제 도로망의 교차점(IP; Intersection Point)를 기준으로 격자(grid)를 나눈다.
- [0015] 여기서 공간을 분할하는 방법은 LUGrid 방법을 기반으로 한다. LUGrid 방법은 도 2 를 참조하여 설명한다. 도 2 는 본 발명의 일실시예에 따른 LUGrid의 전체구조이다.
- [0016] LUGrid는 Grid File을 기반으로 확장한 방법으로 갱신 연산이 삭제와 삽입으로 구성된다는 점에 착안하여 지연 삽입 (Lazy-insertion)과 지연 삭제 (Lazy-deletion) 기법을 적용하여 갱신 비용을 줄이는 기법이다. Grid File과 다른 점은 격자 디렉토리(grid directory)를 디스크가 아니라 메모리에 유지한다는 점인데, Grid File의 격자 디렉토리에 해당하는 LUGrid의 메모리격자(MG:Memory Grid)는 객체의 갱신 정보를 디스크에 바로 저장하지 않고 임시 저장(buffering)한다. 디스크의 버킷 페이지(bucket page)에 해당하는 디스크 격자(DG:Disk Grid)는 이동 객체의 위치 정보를 저장하고 있다. 또한, 해시 기반의 삭제 대상 메모 (MDM:Miss-Deletion Memo)를 유지

하여 삭제 대상이 되어 쓸모가 없게 된 위치 정보들(obsolete entries)을 저장한다. 그리고 MDM에 저장되는 삭제 대상 엔트리의 수를 제한하기 위해 특정 갱신 횟수에 도달하게 되면 Cleaner 수행하여 삭제 대상 위치 정보를 MG와 DG에 반영하게 한다. 도 2 는 이와 같은 정책들이 적용된 LUGrid의 전체적인 구조인데, LUGrid에서 질의를 처리할 경우에는 먼저, 질의 범위(query range)와 MG의 격자 중에 겹침이 발생하는 격자들을 선별한 뒤, MG에 저장되어 있는 최신의 위치 정보 중에 질의 범위에 포함되는 것을 질의 결과 집합에 포함한다. 그리고 선별된 MG의 격자들에 대응되는 DG의 격자들을 한 개씩 읽어오면서 DG의 각 격자에 저장된 이동 객체의 위치 정보가 MDM에서 삭제대상으로 설정되어 있는지와 질의 범위에 포함되는지를 판단하여 질의 결과 집합에 포함하여 질의 처리를 종료한다. 상술한 내용은 통상의 LUGrid 방법과 다르지 않으므로, 자세한 내용은 생략한다.

[0017] 앞서 상술한 바와 같이, 분할부(100)는 도로망의 교차점을 기준으로 공간을 분할한다. 이때, 분할부(100)는 실제 도로망의 교차점(IP; Intersection Point)를 기준으로 격자(grid)를 나눈다. 격자(grid)란 공간 분할시 분할된 하나의 셀(cell)를 의미한다. 이 때, IP는 각 격자당 1개씩 저장되게 하고, 격자의 경계에 포함된 연결점인 CP(Connection point) 역시 격자 정보를 저장하는 메모리 격자(MG;Memory Grid)에 같이 저장된다. 도 3 은 본 발명의 일실시예에 따른 도로네트워크 환경에서의 색인방법 중 도로망 분할과정에 대한 도식도이다. 도 3 을 보면 명확히 알 수 있는데, IP를 기준으로 둘 이상의 IP가 존재하지 않도록 공간을 분할한다.

[0018] S110 단계 이후, 분할부(100)는 분할된 공간마다 식별자를 지정하여 격자로 저장한다(S120).

[0019] 도로 연결 정보를 포함하는 도로 네트워크 공간으로 확장하기 위해 실제 도로망을 분할할 경우에는 Grid File을 사용하여 교차점(IP)를 기준으로 분할하고 연결점(CP)이 포함되게 공간을 분할하는데(S110), Grid File에서는 메인 메모리 상에 해시 테이블과 같은 선형 눈금자(linear scale)를 두어 디스크의 해당 격자의 페이지 아이디(page id)를 저장하여 둔다. 도 4 는 본 발명의 일실시예에 따른 도로 네트워크 환경에서의 도로 공간 분할 화면에 대한 도식도로서, 각각의 격자를 'MG'라고 정의하면, 아래 화면부터 MG1, MG2..로 정의된다. 이때, 제안 색인 구조는 이를 확장하여 메인 메모리 상에 MG에 격자 디렉토리(grid directory)와 도로 연결 정보 및 LUGrid에서 사용하는 속성 값을 저장한다.

[0020] 이와 같이 구성할 경우의 이점은 Grid File의 분할된 격자를 이용하여 도로의 연결성을 확인할 수 있다. 즉, 인접한 격자에 해당 도로가 이어져 있다. 또한, 도로 연결성의 정확한 사용을 위해 격자에 저장된 CP와 인접한 격자의 CP를 사용하여 도로가 연결되었음을 검증할 수 있다. 도 4 를 참고하여 설명하면, MG1의 도로는 MG1에 인접한 격자인 MG2와 MG3의 도로로 연결되어 있기 때문에 그 연결점 CP는 서로 동일하다는 것을 이용하여 MG1.CP1 = MG2.CP2를 만족하고, MG1.CP4 = MG3.CP2인 것으로 도로의 연결성을 검증할 수 있다.

[0021] 분할된 공간인 격자는 IP를 포함하는 격자와 IP 없이 CP만을 포함하는 격자가 있다. 도 4 를 참고하면, MG1은 IP를 포함하는 격자이고, MG2, MG3는 IP를 포함하지 않은 격자가 된다.

[0022] 상기 S120 단계에서 공간이 분할되면, 분할부(100)는 교차점(IP)을 포함하는 격자에 대한 교차점 및 연결점의 정보를 격자 디렉토리에 저장하여 격자색인정보를 생성한다(S130).

[0023] 구체적으로, MG1처럼 IP를 포함하는 격자는 격자 디렉토리에 IP, CP 정보를 포함시킨다. IP를 포함하는 격자의 격자 디렉토리는 도 5a와 같다.

[0024] Grid Info에는 선형눈금자의 정보를 저장하고, 그 옆에 교차점의 위치정보(IP), 총 연결점의 수(CP count) 및 CP count에 저장된 연결점 위치정보(CP-list)를 저장한다.

[0025] 후술하겠으나, 색인갱신처리를 위해서 DG의 디스크 페이지의 수와 디스크 페이지에 포함되는 이동객체의 정보를 추가한다. 이때 저장되는 정보 값은 해당 격자 안에 버퍼링된 갱신 위치 정보 수(N_u), 해당 격자에서 사용 중인 디스크 페이지 수(D_{cnt}), 해당 격자에 연결된 DG 정보(D_{set})를 포함한다. 여기서 D_{set} 은 해당 격자에 연결된 DG의 디스크 페이지 ID(D_{cnt}), 디스크 페이지에 저장된 이동 객체의 위치정보 엔트리 수(N_E), MG 격자안에 버퍼링된 엔트리의 object id, 최신 위치정보($E_1 \sim E_m$)를 포함한다.

[0026] 분할부(100)는 교차점(IP)을 포함하지 않는 격자에 대한 연결점의 정보를 격자 디렉토리에 저장하여 격자색인정

보를 생성한다(S140).

- [0027] 구체적으로, MG2,3처럼 IP를 포함하지 않는 격자는 상기 S130 단계에서 IP정보와 CP Count 정보만 없는 구조이다. IP를 포함하지 않는바, IP 정보가 필요없으며, CP는 2개 밖에 없기 때문에, CP Count 정보는 필요없다. IP를 포함하지 않는 격자의 격자 디렉토리는 도 5b와 같다.
- [0028] 분할부(100)는 삭제 대상 메모(MDM; Miss-Deletion Memo)를 포함한 격자 정보를 격자 디렉토리에 저장한다(S150).
- [0029] 삭제 대상 메모의 격자 디렉토리는 도 5c와 같다. 삭제대상의 식별자정보(OID), 삭제대상 식별자의 위치정보(OLoc), OID에 대한 삭제 대상 수(D_{num}) 정보를 포함한다.
- [0030] S150 단계 이후, 관심객체색인정보 생성부(200)는 도로망에 위치한 관심객체에 대해 정적인 색인을 생성하여 관심객체색인정보를 저장한다(S160).
- [0031] 구체적으로, 건물, 병원과 같은 관심객체(POI ;Position- Of- Interest)에 대해서 색인을 생성한다.
- [0032] 상기 POI는 R*-Tree를 통해 색인을 생성하는데, R*-tree는 공간 객체의 겹침을 최소화하여 공간 검색 성능을 높이는 색인으로, 검색에 주로 사용되고 한 번 구축이 되면 거의 바뀌지 않는 경우 적합한 색인이다. 도 6 은 본 발명의 일실시예에 따른 도로 네트워크 환경에서의 색인방법 중 R*-tree를 이용한 POI 색인방법에 대한 도식도이다. 통상의 R*-tree를 이용한 방법인바, 자세한 설명은 생략한다.
- [0033] 상기 S110 단계 내지 S150 단계에 있어서, 도로망에 대한 색인과 IP, CP와 같은 도로 연결 정보는 고정되어 있고, 거의 변하지 않는다. 또한 색인 구성시에 모든 IP와 CP의 수를 알 수 있으므로, 처음 생성되는 비용을 제외하고는 이동 객체의 갱신 시에 별도의 비용이 필요하지 않다. POI에 대한 색인 역시 고정적인 객체를 대상으로 하고 추가되는 객체가 많지 않으므로 처음 생성되는 비용 외에 거의 필요하지 않는다.
- [0034] 통합부(300)는 상기 분할부(100)를 통해 저장된 격자색인정보와, 관심객체색인정보 생성부(200)를 통해 저장된 관심객체색인정보(POI) 색인정보를 통합한다(S170).
- [0035] 동일 공간을 대상으로 도로망과 POI에 대한 색인인 R*-tree의 단말 노드가 포함된다. 그러므로 도로망 색인의 각 격자에 중첩되거나 포함되는 R*-tree의 단말노드에 대한 포인터를 유지하도록 두 개의 색인정보를 통합한다. 도 7 은 본 발명의 일실시예에 따른 도로 네트워크 환경에서의 색인방법에서의 색인통합 방법에 대한 도식도이다.
- [0036] 도 8 은 본 발명의 일실시예에 따른 도로 네트워크 환경에서의 색인 갱신방법에 대한 흐름도이며, 도 12 는 본 발명의 일실시예에 따른 도로 네트워크 환경에서의 색인 갱신 시스템에 관한 구성도이다.
- [0037] 본 발명의 일실시예에 따른 도로 네트워크 환경에서의 색인 갱신 시스템(S)은, 입력부(100), 검색부(200), 색인 갱신부(300), 메모리 격자 검색부(400), 오버 플로우 생성부(500) 및 위치정보 저장부(600)를 포함하여 이루어진다.
- [0038] 표 1 은 본 발명의 일실시예에 따른 도로 네트워크 환경에서의 색인 갱신방법의 알고리즘이다.

[0039] [표 1] 도로 네트워크 환경에서의 색인 갱신방법의 알고리즘

```

Algorithms 1
input :  $u(oid, loc)$ 
output : buffered  $u$  in grid of MG
1. MG  $m = searchMG(u, oid)$ ;
2. if (  $isNull(m) == false$  ) then
3.   deleteMG( $m$ );  $m.N_u--$ ; usedSlots--;
4. else
5.    $m = searchCoveredMG(u, loc)$ ;
6. endif
7. isfull = true;
8. foreach  $D_{set} \ e$  in  $m.D_{set}$ 
9.   if (  $N_E \neq full$  ) then
10.    insertMG( $u, e.D_{id}$ ); isfull=false;
11.   endif
12. endfor
13. if ( isfull ) then
14.    $d = createOverflowPage()$ ;
15.    $D_{cnt}++$ ; addMG( $d, m$ ); insertMG( $u$ );
16. endif
17.  $u.N_u++$ ; usedSlots++;
18. if (  $u.N_u \geq MaxUpdPerMGCell$  ) or
19.   ( usedSlots  $\geq MaxSlots$  ) then
20.   FlushingUpdates( $m$ );
21. endif

```

[0040]

[0041] 본 발명의 일실시예에 따른 색인 갱신방법은 LUGrid의 연산을 확장하여 기종 기법에서 분할 및 합병 과정을 제거하는데, 분할 및 합병과정을 제거하기 위해서 MG의 격자대응되는 DG의 격자에 오버플로우(overflow) 페이지를 추가할 수 있도록 확장한다. LUGrid의 갱신 연산은 이동객체의 위치정보를 MG(메모리 격자; Memory Grid)에 임시저장(buffering)하는 과정과 임시 저장된 정보를 DG(디스크 격자; Disk Grid)에 쓰는 과정으로 구성된다.

[0042] 먼저, 입력부(100)는 갱신될 새로운 위치정보 $u(OID, Oloc)$ 를 입력받는다(S810).

[0043] 이후, 검색부(200)는 새로운 위치정보가 기존 메모리 격자(MG)에 저장되어 있는지 검색한다(S820).

[0044] S820 단계에서의 검색결과, 상기 새로운 위치정보 u 가 저장되어 있는 경우, 색인 갱신부(300)는 상기 기존 메모리 격자(MG)에 저장되어 있는 위치정보를 삭제하고(S830), 상기 새로운 위치정보 u 를 저장하여 색인을 갱신한다.

[0045] 한편, S820 단계의 검색결과, 저장되어 있지 않을 경우, 메모리 격자 검색부(400)는 상기 새로운 위치정보 u 가 저장될 수 있는 기존 메모리 격자(MG)의 격자를 검색한다(S840)(표 1의 1~6과 같다).

[0046] MG의 격자를 검색하는 방법은 MG의 격자 정보 D_{id} 를 조회하여 저장가능한 DG 격자를 검색한다. 상기 S840 검색결과, DG 모든 페이지에 저장할 공간이 없는 경우, 오버 플로우 생성부(500)는 오버 플로우를 생성하여 저장한다(S850)(표 1의 8~16과 같다).

[0047] 위치정보 저장부(600)는 MG의 격자의 총 엔트리수와 사용슬롯수를 검사하여 DG 격자의 오버 플로우에 새로운

위치정보를 저장한다(S860).

- [0048] 이때, MG의 격자의 총 엔트리수와 사용슬롯수를 검사하여 DG의 격자에 쓰기를 할지를 결정한다(S860)(표 1의 18~21과 같다).
- [0049] MG 내 임의의 격자 한 개에 위치정보를 DG의 격자에 쓰기 위한 알고리즘은 DG의 격자가 여러 페이지에 걸쳐 저장되어 있는 경우는 페이지를 읽기 위해 MG 내 격자 정보 중에서 D_{set} 에 저장되어 있는 여러 페이지 정보를 하나씩 읽으면서 처리하는 것을 제외하고는 기존의 LUGrid 기법과 크게 다르지 않다.
- [0050] MG의 격자에 저장되어 있는 DG의 격자 페이지 정보(D_{set}) 중 한 페이지를 읽어오면서 삭제 대상 메모(MDM)에 저장되어 있는 위치정보를 가져온 DG의 격자페이지에서 삭제한다(S861). 그 후, MG 내 격자에 저장되어 있는 최신 위치 정보를 읽어온 DG의 격자 페이지에 저장(S862)하고, 디스크에 쓰기를 실행한다(S863).
- [0051] 도 9 는 본 발명의 일실시예에 따른 도로 네트워크 환경에서의 색인방법에 대한 범위 질의 처리방법에 대한 흐름도이며, 표 2 는 본 발명의 일실시예에 따른 도로 네트워크 환경에서의 색인방법에 대한 범위 질의(Range query) 처리 알고리즘이다.
- [0052] LUGrid의 범위 질의 처리 알고리즘을 도로의 연결정보를 사용하도록 확장한 것인데, 먼저 질의 영역(Range)과 겹치는 메모리격자(MG)의 집합 Set_{mg} 을 구한다(S910)(표 2의 1~2). 그 후 MG에 포함된 도로망 정보를 사용하여 Set_{mg} 중에 도로가 연결되어 있는 MG인 Set_{input} 을 구한다(S920)(표 2의 3). MG에 저장되어 있는 최신 위치 정보를 결과 집합에 포함시킨 후(S930)(표 2의 4~10), Set_{mg} 에 대응되는 디스크격자(DG)의 페이지를 읽어온다(S940). S940 단계 이후, 읽어온 이동 객체의 위치정보는 IdentifyingEntry 함수를 실행하여 삭제대상메모(MDM)에 표시되어 있는지 검사하여 삭제대상이 아니라면 결과집합에 포함한다(S950)(표 2의 11~18).
- [0053] [표 2] 도로 네트워크 환경에서의 색인방법에 대한 범위 질의(Range query) 처리 알고리즘

```

Algorithms 2
input : range (query range)
output : ResultSet (object set)
1. ResultSet = null;
2.  $Set_{mg} = \text{getOverlapInMG}(\text{range});$ 
3.  $Set_{input} = \text{getRoadConnet}(Set_{mg});$ 
4. foreach MG  $MG_{cell}$  in  $Set_{input}$ 
5.   foreach object obj in  $MG_{cell}$ 
6.     if obj.OLoc  $\in$  range then
7.       ResultSet = ResultSet  $\cup$  obj;
8.     endif
9.   endfor
10. endfor
11. foreach diskpageid  $D_{id}$  in  $Set_{input}$ 
12.   buffer buf = getLoadPage( $D_{id}$ );
13.   foreach object obj in buf
14.     if (obj.OLoc  $\in$  range) and
15.       (IdentifyingEntry(obj)
16.         == CURRENT) then
17.       ResultSet = ResultSet  $\cup$  obj;
18.     endif
19.   endfor
20. endfor
  
```

[0054]

[0055] 도 10 은 본 발명의 일실시예에 따른 도로 네트워크 환경에서의 색인방법에 대한 주변 정보 검색 방법의 흐름도이며, 표 3 는 본 발명의 일실시예에 따른 도로 네트워크 환경에서의 주변 정보 검색 방법의 알고리즘이다.

[0056] [표 3] 도로 네트워크 환경에서의 주변 정보 검색 방법의 알고리즘

```

Algorithms 3
input : range (query range), type
output : ResultSet (object set)
1. ResultSet = null;
2. pointer Setinput = null;
3. Setmg = getOverlapInMG(range);
4. foreach MG MGcell in Setmg
5.   foreach pointer POI in MGcell
6.     Setinput = Setinput ∪ POI
7.   endfor
8. endfor
9. foreach diskpageid Did in Setinput
10.  buffer buf = getLoadPage(Did);
11.  foreach object obj in buf
12.    if (obj.MBR ∩ range) and
13.      (obj.info == type) then
14.      ResultSet = ResultSet ∪ obj;
15.    endif
16.  endfor
17. endfor
    
```

[0057]

[0058] 먼저 질의 영역(Range)과 겹치는 메모리격자(MG)의 집합 Set_{mg}을 구한다(S1010)(표 3의 3). 그 후, Set_{mg}의 값을 한 개씩 가져오면서 해당 격자 MG_{cell}에 저장되어 있는 POI 색인의 단말노드 포인터 정보를 가져온다(S1020)(표 3의 4-8). 그 후, Set_{input}에 저장되어 있는 POI 색인 단말 노드 포인터를 하나씩 가져오면서 단말 노드의 MBR이 질의 영역과 겹치는지 확인하고, 건물, 병원 등의 타입과 비교하여 만족하는 단말 노드를 결과 집합에 포함한다(S1030).

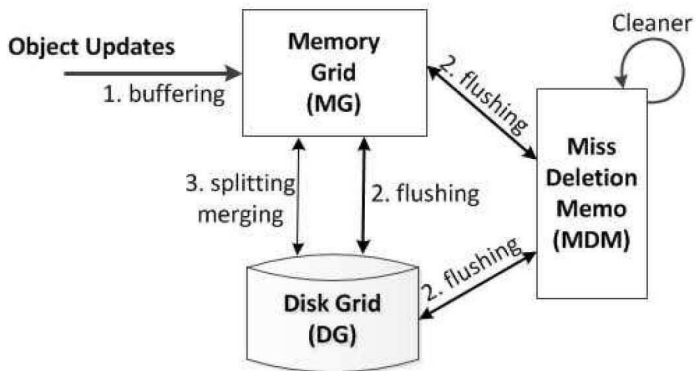
[0059] 이상으로 본 발명의 기술적 사상을 예시하기 위한 바람직한 실시예와 관련하여 설명하고 도시하였지만, 본 발명은 이와 같이 도시되고 설명된 그대로의 구성 및 작용에만 국한되는 것이 아니며, 기술적 사상의 범주를 일탈함이 없이 본 발명에 대해 다수의 변경 및 수정이 가능함을 당업자들은 잘 이해할 수 있을 것이다. 따라서 그러한 모든 적절한 변경 및 수정과 균등물들도 본 발명의 범위에 속하는 것으로 간주되어야 할 것이다.

도면

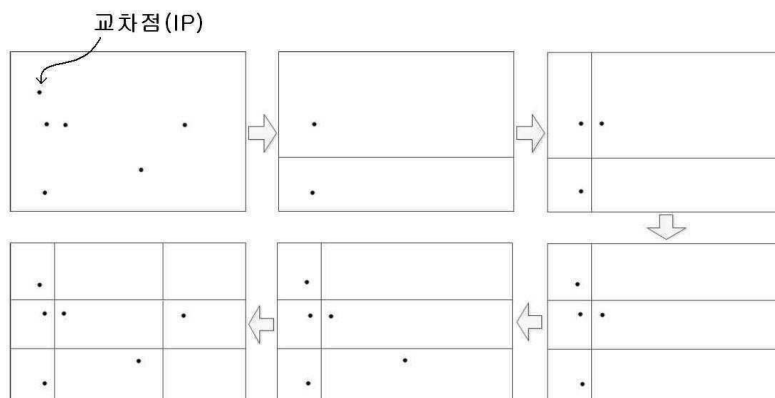
도면1



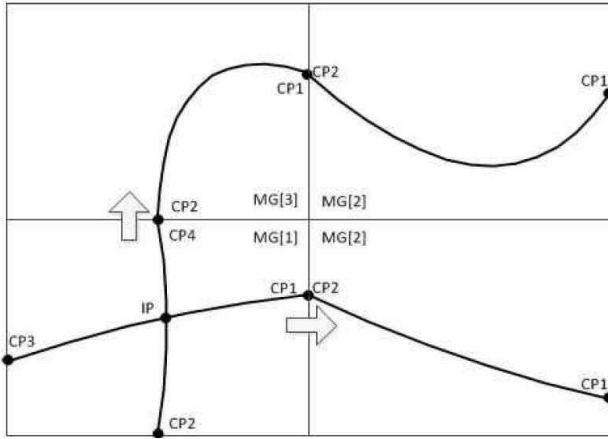
도면2



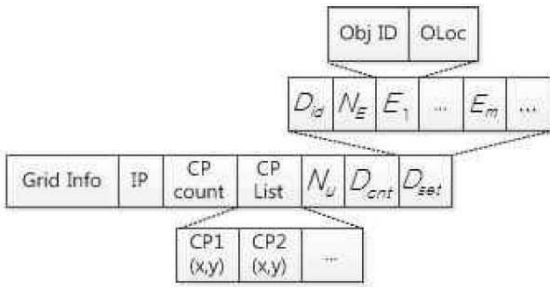
도면3



도면4



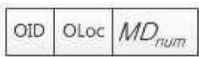
도면5a



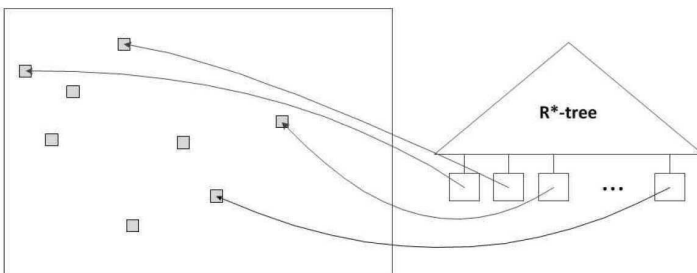
도면5b



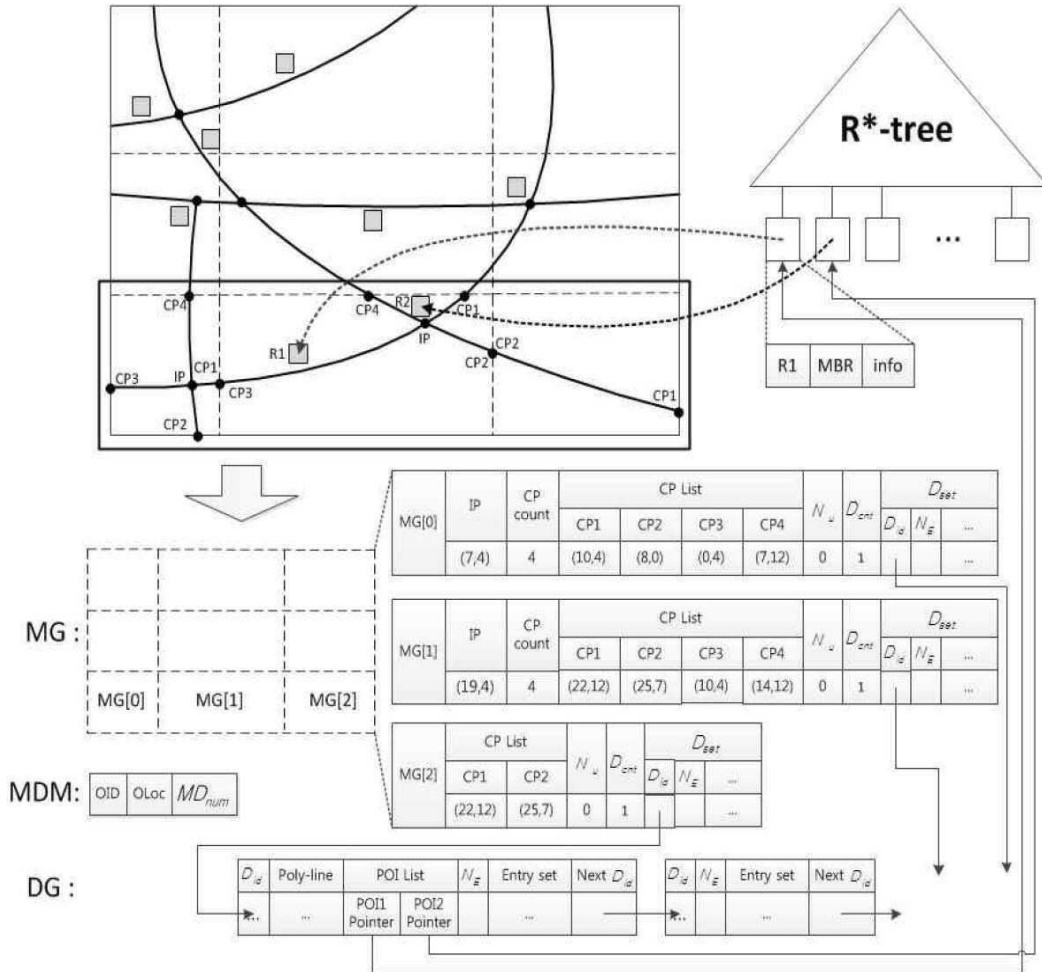
도면5c



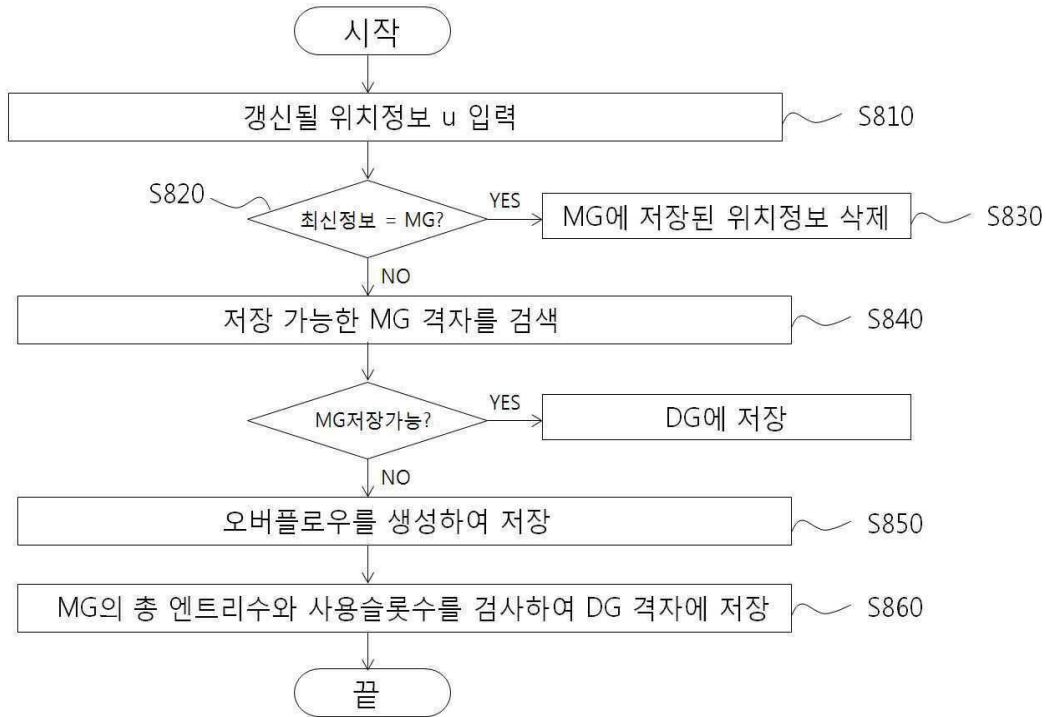
도면6



도면7



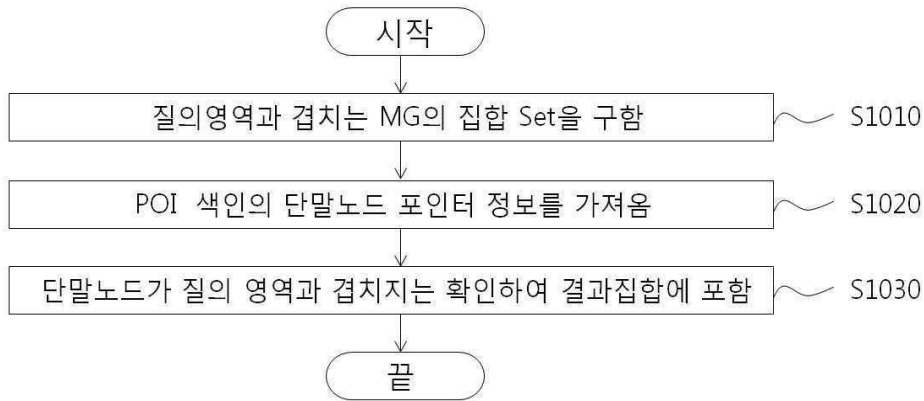
도면8



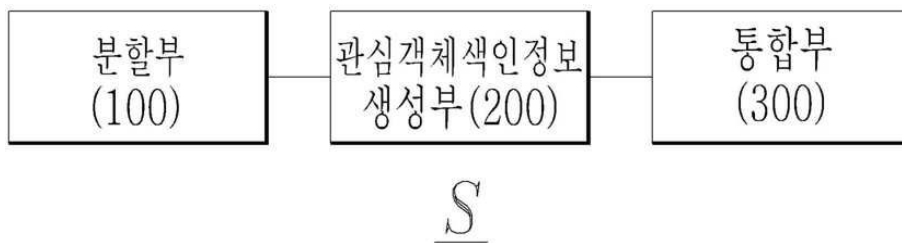
도면9



도면10



도면11



도면12

