



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2012-0026034
(43) 공개일자 2012년03월16일

(51) 국제특허분류(Int. Cl.)
G06F 17/30 (2006.01) G01C 21/26 (2006.01)
(21) 출원번호 10-2011-7024206
(22) 출원일자(국제) 2010년04월19일
심사청구일자 없음
(85) 번역문제출일자 2011년10월14일
(86) 국제출원번호 PCT/EP2010/055128
(87) 국제공개번호 WO 2010/119137
국제공개일자 2010년10월21일
(30) 우선권주장
09158188.4 2009년04월17일
유럽특허청(EPO)(EP)

(71) 출원인
프라운호퍼-게젤샤프트 쉼어 퍼르더롱 데어 안게
반텐 포르숨에.파우.
독일 데-80686 뮌헨 한자스트라세 27체
(72) 발명자
슈람, 안드레아스
독일 베를린 10777, 모츠스트라세 8
에른스트, 티로
독일, 베를린 12557, 쾨펜자일 64아
(74) 대리인
특허법인지명

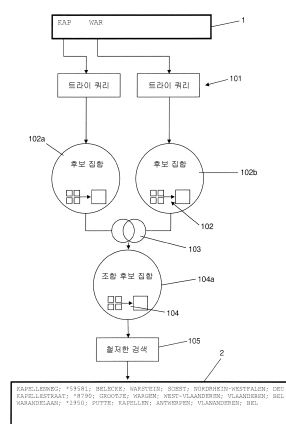
전체 청구항 수 : 총 35 항

(54) 발명의 명칭 데이터베이스 내의 객체들을 검색하는 방법

(57) 요약

다음-레벨 공간적 원소(52)에 대해서 조합 가능한 사전-정의된 수의 공간적 원소들(51)인, 공간을 분할하기 위해 정의되는 공간적 원소들(51)-예컨대 쿼드트리 타일들 또는 옥트리들의 직육면체들-의 컬렉션들(205, 206)에 객체 속성값들을 연관하는 인덱스 데이터 구조(200)에 의해서 데이터베이스 내의 객체들을 검색하는 방법으로서, 제1 입력 검색 패턴을 위해서 상기 인덱스 데이터 구조(200)를 검색하는 단계로서, 상기 제1 입력 검색 패턴이 상기 인덱스 데이터 구조(200)을 통하여 공간적 원소들(51)의 제1 컬렉션(205, 206)에 연관되고, 모든 공간적 원소들(51)을 상기 제1 컬렉션(205, 206)으로부터 공간적 원소들(51)의 제1 후보 집합(102a, 102b) 내에 포함하고, 상기 제1 후보 집합(102a, 102b) 내의 공간적 원소들(51)의 수가 소정의 한계를 초과하면, 공간적 원소들(51)의 일부 또는 모두가 감소된 수의 다음-레벨 공간적 원소들(52)에 조합되는 것인 단계; 제2 입력 검색 패턴을 위해서 상기 인덱스 데이터 구조(200)를 검색하는 단계로서, 상기 제2 입력 검색 패턴이 상기 인덱스 데이터 구조(200)을 통하여 공간적 원소들(51)의 제2 컬렉션(205, 206)에 연관되고, 모든 공간적 원소들(51)을 상기 제2 컬렉션(205, 206)으로부터 공간적 원소들(51)의 제2 후보 집합(102a, 102b) 내에 포함하고, 상기 제2 후보 집합(102a, 102b) 내의 공간적 원소들(51)의 수가 소정의 한계를 초과하면, 공간적 원소들(51)의 일부 또는 모두가 감소된 수의 다음-레벨 공간적 원소들(51, 52)에 조합되는 것인 단계; 상기 제1 후보 집합 및 상기 제2 후보 집합(102a, 102b)로부터, 공간적 원소들(51, 52)의 조합 후보 집합(104a)을 형성하는 단계; 및 결과적 객체들의 집합을 획득하기 위하여, 공간적 원소들(51, 52)의 조합 후보 집합(104a) 내에서, 상기 제1 입력 검색 패턴 및 상기 제2 입력 검색 패턴에 정합하는 객체들을 검색하는 단계를 포함하는 데이터베이스 내의 객체들을 검색하는 방법이 제공된다. 따라서 모바일 내비게이션 장치의 맵 데이터 내에서 객체들을 위해서 자유 텍스트 검색이 허용되는 방법이 제공된다.

대표도 - 도1



특허청구의 범위

청구항 1

특히 2차원 평면 또는 3차원 공간이고, 여기에서는 다음-레벨 공간적 원소(52)에 대해서 조합 가능한[combinable] 사전-정의된(pre-defined) 수의 공간적 원소들(elements)(51)인, 공간을 분할(partition)하기 위해서 정의되는 예컨대 쿼드트리(quadtrees)의 타일들(tiles) 또는 옥트트리(octtree)의 직육면체들인 공간적 원소들[spatial elements](51)의 컬렉션들(collections)(205, 206)에 객체 속성값들[object attribute values]을 연관(associate)하는 인덱스 데이터 구조(200)에 의해서 데이터베이스 내의 객체들을 검색하는 방법으로서,

제1 입력 검색 패턴을 위해서 상기 인덱스 데이터 구조(200)를 검색하는 단계로서, 상기 제1 입력 검색 패턴이 상기 인덱스 데이터 구조(200)를 통하여 공간적 원소들(51)의 제1 컬렉션(205, 206)에 연관되며, 상기 제1 컬렉션(205, 206)으로부터의 모든 공간적 원소들(51)을 공간적 원소들(51)의 제1 후보 집합[candidate set](102a, 102b) 내에 포함하고, 상기 제1 후보 집합(102a, 102b) 내의 공간적 원소들(51)의 수가 소정의 한계(limit)를 초과하면, 공간적 원소들(51)의 일부 또는 모두가 감소된(reduced) 수의 다음-레벨 공간적 원소들(52)에 조합되는 것인 단계;

제2 입력 검색 패턴을 위해서 상기 인덱스 데이터 구조(200)를 검색하는 단계로서, 상기 제2 입력 검색 패턴이 상기 인덱스 데이터 구조(200)를 통하여 공간적 원소들(51)의 제2 컬렉션(205, 206)에 연관되면, 상기 제2 컬렉션(205, 206)으로부터의 모든 공간적 원소들(51)을 공간적 원소들(51)의 제2 후보 집합(102a, 102b) 내에 포함하고, 상기 제2 후보 집합(102a, 102b) 내의 공간적 원소들(51)의 수가 소정의 한계를 초과하면, 공간적 원소들(51)의 일부 또는 모두가 감소된 수의 다음-레벨 공간적 원소들(51, 52)에 조합되는 것인 단계;

상기 제1 후보 집합 및 상기 제2 후보 집합(102a, 102b)로부터, 공간적 원소들(51, 52)의 조합 후보 집합[combined candidate set](104a)을 형성하는 단계; 및

결과적 객체들의 집합을 획득하기 위하여, 공간적 원소들(51, 52)의 조합 후보 집합(104a) 내에서, 상기 제1 입력 검색 패턴 및 상기 제2 입력 검색 패턴에 정합(match)하는 객체들을 검색하는 단계

를 포함하는 데이터베이스 내의 객체들을 검색하는 방법.

청구항 2

제1항에 있어서,

상기 제1 후보 집합 및 상기 제2 후보 집합(102a, 102b)의 교집합(intersection)을 형성하는 것에 의해서, 조합 후보 집합(104a)은 상기 제1 후보 집합 및 상기 제2 후보 집합(102a, 102b)로부터 형성되는 것인 데이터베이스 내의 객체들을 검색하는 방법.

청구항 3

제1항 또는 제2항에 있어서,

상기 인덱스 데이터 구조(200)는 객체 속성값들을 쿼드트리의 공간적 원소들(51)에 연관하고, 각 공간적 원소는 2차원 평면의 포션(portion)에 대응하고, 4개의 인접한(adjointing) 타일들은 다음-레벨 공간적 원소(52)를 묘사하는(representing) 하나의 다음-레벨 타일에 대해서 조합 가능한 것인 데이터베이스 내의 객체들을 검색하는 방법.

청구항 4

제3항에 있어서,

공간적 원소(51)의 조합(combination)을 위해서 공간적 원소(51)의 상기 제2 후보 집합(102a, 102b) 내에서 또는 상기 제1 후보 집합(102a, 102b) 내에서, 4개 또는 그 이하의 인접한 타일들은 하나의 다음-레벨 타일(52)에 대해서 조합되고,

이러한 조합은 다음-레벨 공간적 원소(52)가 조합 공간적 원소들[combined spatial elements](51)의 상위 집합[superset]을 묘사하는 영역(area) 내에서의 안전 컴팩트화[secure compactification]를 묘사하는 것인 데이터

베이스 내의 객체들을 검색하는 방법.

청구항 5

제1항 내지 제4항 중 어느 한 항에 있어서,

특히 텍스트 스트링인 데이터 원소들의 시퀀스를 각각 나타내는 상기 제1 입력 검색 패턴 및 상기 제2 입력 검색 패턴에 대해서, 상기 제1 입력 검색 패턴을 위해서 상기 인덱스 데이터 구조(200)를 검색하는 단계 또는 제2 입력 검색 패턴을 위해서 상기 인덱스 데이터 구조(200)를 검색하는 단계 도중에, 상기 제1 입력 검색 패턴 또는 상기 제2 입력 검색 패턴이 공간적 원소(51, 52)의 객체의 속성값의 프리픽스(prefix)와 정합하는지 검사하는 것인 데이터베이스 내의 객체들을 검색하는 방법.

청구항 6

제1항 내지 제5항 중 어느 한 항에 있어서, 상기 인덱스 데이터 구조(200) 내에서 입력 검색 패턴을 검색하기 위해서, 객체 속성들 또는 객체 속성들의 프리픽스들을, 특히 상기 인덱스 데이터 구조(200)를 통하여, 미리 조립된(assembled) 보조적 후보 집합들에 연관하는 보조적(auxiliary) 인덱스 데이터 구조가 사용되고, 상기 보조적 인덱스 데이터를 검색하여 비공(non-empty) 집합이 산출되면(yield) 이 집합을 원하는 후보 집합(102a, 102b)으로 구성하고, 아니면 원하는 후보 집합(102a, 102b)은 상기 입력 검색 패턴을 위해서 상기 인덱스 데이터 구조(200)를 검색하는 것에 의해서 획득(obtain)하는 것인 데이터베이스 내의 객체들을 검색하는 방법.

청구항 7

제1항 내지 제6항 중 어느 한 항에 있어서,

상기 제1 입력 검색 패턴 및/또는 상기 제2 입력 검색 패턴을 상호 작용적으로[interactively] 입력할 때, 조합 후보 집합(104a)은 증분적으로(incrementally) 감소되는 것인 데이터베이스 내의 객체들을 검색하는 방법.

청구항 8

제1항 내지 제7항 중 어느 한 항에 있어서,

상기 제1 입력 검색 패턴 및/또는 상기 제2 입력 검색 패턴을 상호 작용적으로 입력할 때, 상기 입력 검색 패턴에 의해서 영향을 받는[influenced] 후보 집합(102a, 102b, 104a)의 표현성(expressiveness)에 대한 피드백이 제공되고, 후보 집합(102a, 102b, 104a)의 카디널리티(cardinality)가 감소하면 상기 표현성이 증가하고, 예컨대 후보 집합(102a, 102b, 104a)의 카디널리티가 미리 결정된 임계값보다 작으면 피드백이 제공되고, 및/또는 후보 집합(102a, 102b, 104a)의 카디널리티가 0의 값으로 가정(assume)되면 예러가 지시(indicated)되는 것인 데이터베이스 내의 객체들을 검색하는 방법.

청구항 9

제1항 내지 제8항 중 어느 한 항에 있어서,

공간적 원소들(51, 52)의 조합 후보 집합(104a) 내에서 객체들을 검색하는 것은, 조합 후보 집합(104a)의 카디널리티가 미리 결정된 임계값보다 작아질 때 수행되는 것이고, 결과적 객체들이 출력장치를 통하여 출력되는 것인 데이터베이스 내의 객체들을 검색하는 방법.

청구항 10

제1항 내지 제9항 중 어느 한 항에 있어서,

공간적 원소들(51, 52)의 조합 후보 집합(104a) 내에서 객체들을 검색하는 동안에,

- 이미 검색된 것인 공간적 원소들(51, 52)의 수가 계수(count)되고, 사전-정의된 한계를 넘어설 때 프로세스가 중단(stop)되거나,
- 이미 발견된(found) 객체들의 수가 계수되고, 사전-정의된 한계를 넘어설 때 프로세스가 중단되거나,
- 검색에 대한 경과 시간[elapsed time]이 측정되고, 사전-정의된 한계를 넘어설 때 프로세스가 중단되는 것인 데이터베이스 내의 객체들을 검색하는 방법.

청구항 11

제1항 내지 제10항 중 어느 한 항에 있어서,

특정한 목표 구역[target region]에 연관되는 공간적 원소들(51, 52)만을 고려하여 검색이 제약된(constrained) 것인 데이터베이스 내의 객체들을 검색하는 방법.

청구항 12

제1항 내지 제11항 중 어느 한 항에 있어서,

결과적 객체들의 집합의 객체들이 사용자가 실제로 발견하기를 의도한 객체를 나타내는 개연성(probability)을 지시하는 휴리스틱(heuristic) 규칙들의 집합에 따라서 객체들의 타당성(relevance)이 결정되고, 결과들은 사용자에게 상기 타당성이 감소하는 순으로 표시되는 것인 데이터베이스 내의 객체들을 검색하는 방법.

청구항 13

제1항 내지 제12항 중 어느 한 항에 있어서,

상기 데이터베이스는 내비게이션 장치의 맵(map) 데이터를 포함하는 것이고,

상기 객체들은 상기 맵 데이터의 지리적(geographic) 객체들에 대응하는 것이고, 공간적 원소들(51, 52)에 의해서 분할된 것인 지리적 맵에 대응하는 2차원 평면이 상기 내비게이션 장치에 의해서 표시되는 것인 데이터베이스 내의 객체들을 검색하는 방법.

청구항 14

제13항에 있어서,

상기 제1 입력 검색 패턴 및/또는 상기 제2 입력 검색 패턴을 상호 작용적으로 입력할 때, 특히 상기 상호 작용적인 입력 동안에 증분적인 방식으로, 공간적 원소들(51, 52)의 조합 집합[combined set]에 대응하는 지리적 영역이 출력되는 것인 데이터베이스 내의 객체들을 검색하는 방법.

청구항 15

제1항 내지 제14항 중 어느 한 항에 있어서,

상기 제1 입력 검색 패턴 및/또는 상기 제2 입력 검색 패턴은, 특히 복수의 문자들 및/또는 숫자들이 키패드(3)의 하나의 키에 할당(assign)되어 있는 숫자 키패드(3)인 간략화 키패드[reduced keypad](3)를 통하여 입력되는 것이고, 각각의 키는 상기 입력 검색 패턴의 문자 당 단 한번만 눌러지는 것인 데이터베이스 내의 객체들을 검색하는 방법.

청구항 16

제1항 내지 제15항 중 어느 한 항에 있어서,

상기 제1 입력 검색 패턴 및/또는 상기 제2 입력 검색 패턴은, 휴리스틱하게(heuristically) 해석(interpret)되어, 특정한 사전-정의된 역할들(roles)의 선택을 각각 가지는 것이고, 여기에서 역할은 개별적인(respective) 입력 검색 패턴의 가능한 근원적인(underlying) 의미를 지시하는 것인 데이터베이스 내의 객체들을 검색하는 방법.

청구항 17

제16항에 있어서, 상기 제1 입력 검색 패턴 및/또는 상기 제2 입력 검색 패턴을 휴리스틱하게 해석하는 것에 의해서, 역할-특화(role-specific) 휴리스틱 후보 집합(102a, 102b)이 각각의 입력 검색 패턴에 대해서 연산(compute)되고, 역할-특화 휴리스틱 후보 집합(102a, 102b)을 해석하는 것에 의해서, 검색 쿼리(query)의 특정한 해석을 위해서 조합 휴리스틱 후보 집합[combined heuristic candidate set](104a)이 획득되는 것인 데이터베이스 내의 객체들을 검색하는 방법.

청구항 18

제17항에 있어서,

각각의 경우에서의 검색 쿼리의 여러 개의 서로 다른 해석을 위한 제1 스텝(step)에서는, 조합 휴리스틱 후보 집합(104a)이 연산되고 철저하게(exhaustively) 검색되고, 제2 스텝에서는, 상기 검색 쿼리를 특정한 해석에 제약(constraining)하지 않는 조합 후보 집합(104a)이 연산되고 철저하게 검색되는 것인 데이터베이스 내의 객체들을 검색하는 방법.

청구항 19

제18항에 있어서,

상기 연관된 역할-특화 휴리스틱 후보 집합들(102a, 102b) 및 조합 휴리스틱 후보 집합들(104a)을 연산하지 않으면서 서로 다른 해석들이 우선 상징적으로(symbolically) 열거되고(enumerate), 하나 이상의 상기 역할-특화 휴리스틱 후보 집합들(102a, 102b)이 상기 인덱스 데이터 구조(200) 내에서 룩업(look-up)을 통하여 비어 있는[empty] 것으로 검출(detected)된 것에 대해서, 이러한 해석들이 더 이상 고려(consider)되지 않는 것인 데이터베이스 내의 객체들을 검색하는 방법.

청구항 20

제18항 또는 제19항에 있어서,

입력 검색 패턴을 위해서 고려되는 서로 다른 해석들의 선택 또는 서로 다른 해석들의 상징적 열거는, 상기 인덱스 데이터 구조(200) 내에서 룩업을 통하여 획득되는 하나 이상의 역할-특화 휴리스틱 후보 집합들(102a, 102b, 104b)의 예측된(estimated) 카디널리티에 따라서 결정되는 것인 데이터베이스 내의 객체들을 검색하는 방법.

청구항 21

예컨대 쿼트리의 타일들 또는 옥트트리의 직육면체들인 원소들(51)의 컬렉션들(205, 206)에 데이터베이스 객체 속성값들을 연관하는 인덱스 데이터 구조(200)에 의해서 데이터베이스 내의 객체들을 검색하는 방법으로서,

입력 검색 쿼리 내에서, 휴리스틱하게 제1 입력 검색 패턴 및/또는 제2 입력 검색 패턴을 각각이 특정한 사전-정의된 역할들의 선택을 가지도록 해석하는 단계로서, 여기서 역할은 객체 속성값에 대한 개별적인 입력 검색 패턴의 가능한 근원적인 의미를 지시하는 것인 단계;

휴리스틱하게 상기 제1 입력 검색 패턴 및/또는 상기 제2 입력 검색 패턴을 해석하는 것에 의해서, 각각의 입력 검색 패턴을 위해서 상기 인덱스 데이터 구조에 의해서 원소들(51)의 역할-특화 휴리스틱 후보 집합(102a, 102b)을 연산하는 단계로서, 상기 역할-특화 휴리스틱 후보 집합(102a, 102b)은 미리-기술된(pre-scribed) 역할 내에서 상기 입력 검색 패턴을 정합하는 적어도 하나의 속성값을 포함하는 원소들(51)의 컬렉션들을 포함하는 것인 단계;

상기 검색 쿼리의 특정한 해석을 위해서 조합 휴리스틱 후보 집합(104a)을 획득하도록 상기 제1 입력 검색 패턴 및 상기 제2 입력 검색 패턴을 위해서 상기 역할-특화 휴리스틱 후보 집합(102a, 102b)을 교차(intersect)하는 단계; 및

원소들(51)의 상기 조합 휴리스틱 후보 집합(104a) 내에서, 상기 제1 입력 검색 패턴 및 상기 제2 입력 검색 패턴에 정합하는 객체들을 식별하여, 결과적 객체들의 집합을 획득하는 단계;

를 포함하는 데이터베이스 내의 객체들을 검색하는 방법.

청구항 22

제21항에 있어서,

각각의 경우에서의 검색 쿼리의 여러 개의 서로 다른 해석을 위한 제1 스텝(step)에서는, 조합 휴리스틱 후보 집합(104a)이 연산되고 필요시 철저하게 검색되고, 제2 스텝에서는, 상기 검색 쿼리를 특정한 해석에 제약하지 않는 조합 후보 집합(104a)이 연산되고, 필요시 철저하게 검색되는 것인 데이터베이스 내의 객체들을 검색하는 방법.

청구항 23

제21항 내지 제22항 중 어느 한 항에 있어서,

보조적 인덱스 데이터 구조가 적용되어(employed) 특정한 입력 검색 패턴들을 위한 사전-컴팩트화된(pre-compactified) 후보 집합들(102a, 102b)을 제공하고, 주요(main) 인덱스 데이터 구조(200)가 적용되기 전에 상기 보조적 인덱스 데이터 구조는 항상 검사(probe)되는 것인 데이터베이스 내의 객체들을 검색하는 방법.

청구항 24

제21항 내지 제23항 중 어느 한 항에 있어서,

주어진 입력 검색 패턴에 대해서 우선 연산된 후에 상기 후보 집합들(102a, 102b) 및 조합 후보 집합들(104a)은 캐시(cache) 메모리 데이터베이스 내에 저장되는 것인 데이터베이스 내의 객체들을 검색하는 방법.

청구항 25

제21항 내지 제24항 중 어느 한 항에 있어서,

상기 인덱스 데이터 구조(200)는 객체 속성의 정확한 값들 뿐만 아니라 추가적으로 이러한 값들의 사전-정의된 변화량들(variations)을 위하여 엔트리들(entries)을 가지는 것이고, 이는 특정한 문자들을 다른 문자들로 대체하거나, 또는 특정한 문자들을 원래의 값에 부가하거나, 또는 원래의 값에서 특정한 문자를 생략(omit)하거나, 또는 원래의 값 내에서 위치들에 대해서 일부의 문자들의 위치들을 교환(exchange)하는 것인 데이터베이스 내의 객체들을 검색하는 방법.

청구항 26

제21항 내지 제25항 중 어느 한 항에 있어서,

상기 제1 및 상기 제2 입력 검색 패턴을 위한 제1 및 제2 후보 집합(102a, 102b)은 공간적 원소들(51)을 상기 제1 및 제2 후보 집합(102a, 102b) 내로 포함하는 것에 의해서 확장(extend)되는 것이고, 상기 제1 및 제2 후보 집합(102a, 102b)이 조합되기 전에, 상기 제1 및 제2 후보 집합(102a, 102b) 내에 이미 존재하는 공간적 원소들(51)에 지리적으로 인접하는 것인 데이터베이스 내의 객체들을 검색하는 방법.

청구항 27

제21항 내지 제26항 중 어느 한 항에 있어서,

상기 제1 및 상기 제2 입력 검색 패턴은 음성 입력 장치와 같은 매체를 통하여 입력되고, 단일 검색 쿼리 대신에 여러 개의 입력 서치 패턴들을 각각 구성하는 다수의 검색 쿼리 변형들(variants)이 산출되고,

인덱스 데이터 구조(200) 내의 룩업을 통하여 검색 쿼리 변형들로부터 검색 패턴을 위해서 연산된 후보 집합들(102a, 102b)의 상기 예측된 카디널리티 또는 검색 쿼리 변형들을 위해 연산된 조합 후보 집합들(104a)의 카디널리티가 사용되어, 추가적 처리[further processing]를 위한 검색 쿼리 변형들의 부분 집합[subset]의 선택을 유도(guide)하거나 추가적 처리를 위한 순서(order)에 영향을 미치는 개별적인 검색 쿼리 변형들에 대한 우선 순위들[priorities]을 할당하는 것을 유도하는 것인 데이터베이스 내의 객체들을 검색하는 방법.

청구항 28

제21항 내지 제27항 중 어느 한 항에 있어서,

인덱스 데이터 구조(200) 내의 룩업을 통하여 획득되는 것인 하나 이상의 후보 집합들(102a, 102b, 104a)의 예측된 카디널리티는, 입력 검색 패턴들의 다수 또는 입력 검색 패턴의 표현성(expressivity)을 측정하는데 사용되는 것인 데이터베이스 내의 객체들을 검색하는 방법.

청구항 29

제21항 내지 제28항 중 어느 한 항에 있어서,

미래의(future) 검색 프로세스들을 위해서 사용되도록, 검색 결과의 사용자의 최종 선택이 지속성(persistent) 저장 매체에 등록되는 것인 데이터베이스 내의 객체들을 검색하는 방법.

청구항 30

제21항 내지 제29항 중 어느 한 항에 있어서,

공간적 원소들(51, 52)의 조합 후보 집합의 철저한 검색의 결과들로서,

입력 검색 쿼리의 모든 입력 검색 패턴들에 정합하는 속성을 가지는 객체들이 등록되는 것 뿐만 아니라, 객체 속성에 정합하지 않는 하나 이상의 입력 검색 패턴들에 대한 객체들도 등록되는 것인 데이터베이스 내의 객체들을 검색하는 방법.

청구항 31

제21항 내지 제30항 중 어느 한 항에 있어서,

공간적 원소들(51)의 연관된 컬렉션들(204-206) 및 인덱스 데이터 구조(200)는, 상기 인덱스 데이터 구조(200)를 사용하여 실제로 검색되는 데이터베이스로부터 생성(generate)되는 것이 아니고, 실제 검색되는 데이터베이스의 모든 또는 대부분의 객체들을 포함하는 더 큰 데이터베이스로부터 생성되는 것인 데이터베이스 내의 객체들을 검색하는 방법.

청구항 32

제21항 내지 제31항 중 어느 한 항에 있어서,

인덱스 데이터 구조(200) 내에서 주어진 단어의 존재(presence) 또는 부존재(absence), 또는 인덱스 데이터 구조(200) 내에서 록업을 통하여 획득되는 하나 이상의 연관된 후보 집합들(102a, 102b)의 예측된 카디널리티, 또는 하나 이상의 조합 후보 집합들(104a)이 사용되어, 잘못 타이핑된 검색 패턴들을 보정 및/또는 식별하거나, 및/또는, 잘못 타이핑된 검색 패턴들을 정확한 검색 패턴으로 변환하도록 의도하는 자동적으로 생성된 사용자-특화 검색 패턴의 수정들(modifications)을 평가(evaluate)하는 것인 데이터베이스 내의 객체들을 검색하는 방법.

청구항 33

제21항 내지 제32항 중 어느 한 항에 있어서, 합성(synthetic) 공간 위치 값이, 비-공간적(non-spatial) 데이터베이스의 검색을 가능하게 하기 위해서 기존의(existing) 데이터베이스 객체 속성값들의 기능(function)으로서, 각 데이터베이스 객체에 연관되는 것인 데이터베이스 내의 객체들을 검색하는 방법.

청구항 34

제21항 내지 제33항 중 어느 한 항에 있어서,

상기 방법을 실행하도록 구성되고 서버들로서 작용하는 하나 이상의 컴퓨팅 장치들과 클라이언트들로서 작용하는 하나 이상의 컴퓨팅 장치들을 포함하는 시스템을 사용하고, 각 클라이언트는 사용자 입력으로서 검색 쿼리들을 수용(accept)하도록 및/또는 검색 결과들을 하나 이상의 사용자들에게 다시 통신하도록 구성되고, 각 클라이언트는 영구적으로 또는 간헐적으로(intermittently) 인터넷과 같은 네트워크를 통하여 또는 무선 또는 적외선 통신과 같은 단방향 또는 양방향 통신 링크를 통하여 서버들에 연결되는 것인 데이터베이스 내의 객체들을 검색하는 방법.

청구항 35

제1항 내지 제34항 중 어느 한 항에 기재된 방법을 실행하도록 구성되며, 내비게이션 장치와 같은, 특히 모바일 컴퓨팅 장치인, 컴퓨팅 장치.

명세서**기술 분야**

본 발명은 데이터베이스 내의 객체들을 검색하는 방법에 관한 것이고, 특히 모바일(내비게이션) 장치 상의 맵(map) 검색에 관한 것이다.

[0001]

배경 기술

- [0002] GPS-기반 모바일 내비게이션 시스템은, 특히 소위 개인 내비게이션 장치[personal navigation devices, PND] 또는 점점 증가하는 휴대 전화기, 특히 스마트폰 상의 소프트웨어 내비게이션 애플리케이션은, 요즘 자리를 확실히 잡고 있다[well established]. 이하 이들을 전체적으로 내비게이션 장치라고 지칭한다.
- [0003] 비록 스마트폰 세그먼트(segment) 내에서의 증가하는 네트워크형(networked) 내비게이션 솔루션들[소위 커넥티드(connected) 내비게이션; 예컨대 인터넷-기반 맵 서비스를 위한 모바일 클라이언트들]이 역할을 하고 있지만, 내비게이션 솔루션들의 대다수는 자족(autarkic) 방식으로 동작하도록 설계되어, 영구적이거나 또는 빈번한 네트워크 접속을 요구하지 않는다(소위 오프라인 내비게이션). 이러한 방식은 한편으로는 네트워크 연결들의 높은 비용 때문에 장점을 가지지만, 다른 한편으로는 특히 여행 중에는, 해당 국가의 (시골) 지역들에서의 제한된 커버리지 또는 서로 다른 국가들에서의 서로 다른 네트워크 설정들(setups) 때문에, 네트워크 연결들이 항상 가능한 것은 아니다.
- [0004] 자족 모바일 내비게이션 장치는 전체 맵 자료(material)를 포함하여야 하며, 통상적으로는 매우 압축된 형태로 플래쉬 메모리 카드(예컨대 SD 카드)와 같은 데이터 저장 매체 상에 포함한다. 내비게이션 장치 내에서, 사용자 상호 작용[interaction] 조작들(operations) 중 중요한 것의 하나는, 예컨대 거리(street) 주소 또는 특정한 관심 지점(POI, Point of interest), 예컨대 사용자에게 내비게이션 경로(route)의 목적지(destination)를 한정하기 위한, 맵 내에서의 임의의 객체에 대한 입력 및 검색이다. 여기에서 특히 필요로 하는 키 스트로크들(strokes)의 수, 가능한 대기 시간, 상호 작용 피드백 등의 입력 조작의 복잡성(complexity)은, 사용자의 만족도와 시스템의 사용 가능성을 결정한다.
- [0005] 종래의 내비게이션 장치는 일반적으로 맵 내에서 객체들을 발견(find)하고 위치 배치(locate)하기 위해서, 두 가지의 가능성을 제시한다. 한편으로는, 사용자는 모바일 내비게이션 장치 상에 디스플레이되는 맵 세그먼트를 움직이고[패닝(panning)] 주밍(zooming)하여 목적지를 결정하고, 따라서 목적지를 포함하는 맵 구역(region)에 도달한다. 목적지는 [예컨대 터치-민감(sensitive) 스크린] 포인터를 가지고 목적지를 택하는(고르는) 것에 의해서 결정될 수 있다. 이러한 프로세스는, 그러나 지루한 것이고, (모바일) 내비게이션 장치의 하드웨어 한계들에 의해서 대기 시간이 발생할 수 있다. 또한, 사용자가 맵 내에서 지리적인 목적지를 발견하기 위한 어느 정도 정확한 지식을 미리 가지고 있어야 하나 종종 그렇지 않다는 점에서, 그러한 프로세스는 종종 실용적이지 않다.
- [0006] 따라서, 다른 한편으로, 내비게이션 장치들은 통상적으로, (예컨대 거리 명칭 또는 관심 지점의 명칭인) 그 명칭에 따라서 사용자가 객체를 위치 배치할 수 있도록, 텍스트-기반 검색 기능을 제공한다.
- [0007] 데스크탑 컴퓨터나 서버와 같은 고정된 컴퓨터 상에서는, 모든 사용 가능한 맵 자료의 전체 저장된 데이터 레코드를 철저하게(exhaustively) 검색하는 것은 문제가 되지 않는다. 따라서 이러한 검색 기능은 다수의 지리-정보 시스템들(GIS) 및 지리-데이터베이스 소프트웨어 제품들에서 지원되고 있다.
- [0008] 인터넷-기반 맵 서비스들(예컨대 요즘의 구글 맵스)은 사용자 친화[user friendliness] 관점에서 검색 기능들을 최적화하고 있고, 그 내부에서, 인터넷 검색 엔진들로부터 알려진 기능들을 사용한다. 이러한 것은 맵 내의 모든 객체들의 모든 속성들과 사용자에게 의해서 입력된 다중 검색 용어들(terms)을 비교하는 자유 텍스트 검색[free text search]을 허용한다. 검색 결과들은 이후 그들의 타당성(relevance)에 따라서 정렬(sort)되고 표시된다. 그러나 이러한 솔루션들 및 기본적인 알고리즘들은, 모바일 내비게이션 장치들의 주 메모리 및 프로세스 모두가 그러한 소프트웨어 응용들을 실행하기 위해서 충분하지 않기 때문에, 종래의 모바일 내비게이션 장치들에는 적용될 수 없다. 또한 그러한 내비게이션 장치들의 메모리는, (압축되지 않았거나 단지 근소하게 압축된 것인) 필요로 하는 표준화된 맵 포맷들(formats)을 보유(hold)하기에는 그리고 소프트웨어의 실행에 필요하며 데이터베이스 검색 도중에 생성되는 추가적인 데이터 파일들(예컨대 데이터베이스 인덱스 파일들)을 저장하기에는, 충분한 메모리 공간을 제공하지 않는다.
- [0009] 이러한 제한들 때문에, 종래의 모바일 내비게이션 장치들은 텍스트-기반 검색을 위해서 단지 단순한 옵션들만을 제공한다. 검색 프로세스는, 여기서는, 일반적으로, 사용자가 우선 (예컨대 국가 및/또는 도시와 같은) 구역을 그리고 최종적으로 거리 주소 또는 다른 검색 용어(예컨대 관심 지점의 명칭) 선택하는 것인, 계층적(hierarchical) 지리적 국한(confinement)을 요구한다. 일반적인 모바일 내비게이션 장치들의 현존하는 하드웨어 한계들 때문에 종래의 검색 알고리즘들이 인터넷-기반 맵 서비스들의 방식과 같은 모든 구역들에 대한 자유 텍스트 검색을 허용하지 않기 때문에, 종래의 검색 알고리즘들은 구역을 입력한 다음에만, 가능한 도시 주소들 또는 다른 검색 용어들에 대한 검색이 수행된다.

발명의 내용

해결하려는 과제

[0010] 본 발명의 목적은 모바일 내비게이션 장치의 맵 데이터 내에서 객체들의 자유 텍스트 검색을 허용하는 방법 및 내비게이션 장치를 제공하는 것을 목적으로 한다.

과제의 해결 수단

[0011] 본 발명의 목적은 청구항 제1항의 특징을 포함하는 방법에 의해서 달성된다.

[0012] 이에 따르면, 인덱스 데이터 구조에 의해서 데이터베이스 내의 객체들을 검색하는 방법이 제공된다. 객체들 각각은, 여기서는, 2차원 또는 3차원 공간적 로케이션(location)을 가지며, 검색 패턴에 대해서 정합(match)될 수 있다. 인덱스 데이터 구조는, 예컨대 객체들의 명칭들과 같은 객체 속성값들을, 특히 2차원 평면 또는 3차원 공간인 공간을 분할(partition)하기 위해서 정의되는 예컨대 쿼드트리(quadtrees)의 타일들(tiles) 또는 옥트트리(octree)의 직육면체들인 공간적 원소들[spatial elements]의 컬렉션들(collections)에 연관(associate)한다. 사전-정의된(pre-defined) 수의 공간적 원소들은 다음-레벨 공간적 원소에 대해서 조합 가능[combinable]하다.

[0013] 본 방법에서, 인덱스 데이터 구조는 제1 입력 검색 패턴을 위해서 검색되며, 제1 입력 검색 패턴이 인덱스 데이터 구조를 통하여 공간적 원소들의 제1 컬렉션에 연관되면, 제1 컬렉션으로부터의 모든 공간적 원소들은 공간적 원소들의 제1 후보 집합[candidate set] 내에 추가된다. 여기에서, 제1 후보 집합 내의 공간적 원소들의 수가 소정의 최댓값을 초과하면, 공간적 원소들의 일부 또는 모두가 감소된(reduced) 수의 다음-레벨 공간적 원소들에 조합된다.

[0014] 제1 입력 검색 패턴을 위한 인덱스 데이터 구조의 검색에 후속하여 또는 병행하여, 인덱스 데이터 구조는 제2 입력 검색 패턴을 위해서 검색되며, 제2 입력 검색 패턴이 인덱스 데이터 구조를 통하여 공간적 원소들의 제2 컬렉션에 연관되면, 제2 컬렉션으로부터의 모든 공간적 원소들은 공간적 원소들의 제2 후보 집합 내에 추가된다. 여기에서, 제2 후보 집합 내의 공간적 원소들의 수가 소정의 최댓값을 초과하면, 공간적 원소들의 일부 또는 모두가 감소된 수의 다음-레벨 공간적 원소들에 조합된다.

[0015] 제1 후보 집합 및 제2 후보 집합이 획득되면, 제1 후보 집합 및 제2 후보 집합의 교집합(intersection)이 형성되어 공간적 원소들의 조합 후보 집합[combined candidate set]을 획득한다. 제1 입력 검색 패턴 및 제2 입력 검색 패턴과 정합하는 객체들을 위해서, 이 조합 후보 집합은 철저한 방식으로 이후 검색되어, 모든 입력 검색 패턴들에 정합하는 결과적(result) 객체들의 집합을 결국(eventually) 획득한다.

[0016] 본 발명은 후보 집합들의 매우 효율적인 연산(computing)을 허용한다. 특히 비표현적(inexpressive) 검색 패턴들(예컨대 매우 많은 객체들에서 발생하는 짧은 검색 단어들)에 대해서, 종래의 방법들을 사용하여 획득되는 후보 집합들은 매우 크며, 또한 인-코어(in-core) 데이터 구조들로서의 그들의 묘사(representation) 및 그들의 조합된 처리(processing)(예컨대 검색 프로세스 동안에 그들의 교집합을 연산하는 것)은 (예컨대 모바일 내비게이션 장치들과 같은) 제약된(constrained) 자원들을 가지는 하드웨어 플랫폼들 상에서는 실용적이지 않다. 본 발명은 작은 양의 메모리만을 사용하여 공간적 원소들의 큰 후보 집합들을 묘사하는 것과 후보 집합들을 작은 연산 노력만으로 처리하는 것을 허용한다.

[0017] 두 개의 후보 집합들 각각이 매우 큰 경우에서도, 그들의 교집합은 비교적으로 작은 카디널리티(cardinality)를 가질 수 있으며, 결과적인 조합 후보 집합은 효율적으로 결정될 뿐만 아니라, 또한 모든 입력 검색 패턴들에 정합하는 결과적 객체들을 위해서 철저하게 검색될 수 있을 정도로 작다. 실제로, 본 발명에 따르면, 상호 작용 사용자 입력의 초기 단계들에서 발생하는 것과 같은 매우 비표현적 검색 패턴들에 대해서도 효율적으로 검색 결과들을 결정하는 것이, 제약된 하드웨어 플랫폼들 상에서도 가능하다.

[0018] 이 문맥(context) 내에서, 입력 검색 패턴은, 사용자에게 의해서 상호 작용적으로 입력될 수 있고 알파벳 문자들 및/또는 숫자들을 포함하는 텍스트 스트링일 수 있는 패턴이다.

[0019] 객체들을 검색하는 목적을 위해서, 본 발명은 하나 또는 두 개의 입력 검색 패턴에 한정되는 것은 아니다. 오히려, 세 개 또는 그 이상의 검색 패턴들이 사용자에게 의해서 입력될 수 있고, 집합 이론 법칙[the laws of set theory]에 의해서 용인되는(permitted) 임의의 순서(order)에 따라서 검색 패턴들과 연관된 모든 후보 집합들을 교차(intersect)하는 것에 의해서 조합 후보 집합이 결정될 수 있다.

- [0020] 공간적 원소들의 컬렉션은, 객체의 명칭과 같은, 객체의 속성값에 연관되는 하나 또는 다수의 공간적 원소들을 포함하는 집합이다.
- [0021] 인덱스 데이터 구조는, 객체 속성값들을, 예컨대 2차원 평면을 분할하기 위해서 사용되는 쿼드트리(quadtree)의 공간적 원소들의 컬렉션들과 연관한다. 쿼드트리 내에서, 각각의 공간적 원소는 타일(tile), 즉 2차원 평면의 직사각형 세그먼트에 대응하고, 여기에서 4개의 인접한(adjacent) 타일들은, 4개의 원래 타일들을 커버(cover)하는 영역을 가지며 다음-레벨 공간적 원소(52)를 나타내는 하나의 다음-레벨 타일에 대해서 조합될 수 있다. 일반적으로 쿼드트리는 모든 넌리프 노드(non-leaf node)가 4개의 칠드런(children)을 가지는 트리로서 정의된다. 여기에서는, 균일한 깊이의 쿼드트리가 적용될 수도 있다.
- [0022] 인덱스 데이터 구조는 트라이(trie)라고 지칭되는 구조를 가질 수 있다. 트라이는 그 에지들(edges)이 비공(non-empty) 스트링들로 라벨링(label)되고 그 노드들이 루트에서 노드까지의 길(path)의 에지 라벨들의 연속(concatenation)에 의해서 라벨링된다. 루트 자체는, 예컨대 비어 있는(empty) 스트링으로 라벨링된다.
- [0023] 공간적 원소들의 제2 후보 집합 내에 또는 제1 후보 집합 내에 있는 공간적 원소들의 조합을 위해서, 4개 또는 그 이하의 인접한 타일들은 하나의 다음-레벨 타일에 조합되고, 이러한 조합은 다음-레벨 공간적 원소가 조합 공간적 원소들의 [적절한(proper) 또는 부적절한(improper)] 상위 집합[superset]을 묘사하는 영역(area) 내에서의 안전 컴팩트화[secure compactification]를 묘사하는 것이다.
- [0024] 제1 입력 검색 패턴 또는 제2 입력 검색 패턴을 위해서 인덱스 데이터 구조를 검색하는 것에 의해서, 제1 입력 검색 패턴 또는 제2 입력 검색 패턴이 공간적 원소의 객체 속성에 정합하는지 여부를 검사한다. 이를 위해서, 각각의 검색 패턴은 알파벳 문자들 및/또는 숫자들에 의해서 형성되는 스트링인 텍스트 스트링을 나타낼 수 있다. 텍스트 스트링은, 검색 도중에, 객체의 속성들(예컨대 명칭)과 비교된다. 만약 텍스트 스트링이 속성이나 속성의 프레픽스(prefix)와 정합하면(즉 속성 명칭의 시작이 텍스트 스트링과 정합하는 것), 공간적 원소들의 연관된 컬렉션은 개별적인 후보 집합들에 추가된다. 다른 말로 하면, 검색 패턴에 대해서 정합하는 프레픽스를 가지는 (즉, 인덱스 키들이 검색 패턴 내에서 포함된 문자들 또는 숫자들로 시작하는) 모든 인덱스 키들(keys)에 대해서 검색하는 것에 의해서, 인덱스 데이터 구조는 입력 검색 패턴을 위해서 검색된다. 이러한 모든 인덱스 키들에 대해서, 인덱스 데이터 구조를 통하여 공간적 원소들의 컬렉션에 인덱스 키가 연관된다면, 이러한 컬렉션으로부터의 모든 공간적 원소들은 공간적 원소들의 후보 집합에 추가되고, 만약 후보 집합 내의 공간적 원소들의 수가 미리 결정된 최대값을 넘으면, 공간적 원소들의 일부 또는 전부가 조합되어 다음-레벨 공간적 원소들의 감소된 수로 된다.
- [0025] 이러한 기술에 의해서, 검색 패턴에 정합하는 인덱스 데이터 구조 내의 모든 키들에 연관된 모든 후보 집합들의 컴팩트화된 집합 유니온(union)은 증분적으로(incrementally) 그리고 효율적으로 작성되어, 제한된 크기의 관리 가능한 메모리 묘사를 가진다.
- [0026] 본 발명의 일 실시예에서는, 인덱스 데이터 구조 내에서 입력 검색 패턴을 검색하기 위해서, 객체 속성값들 또는 객체 속성값들의 프레픽스들을, 인덱스 데이터 구조를 통하여, 미리 조립된 보조적 후보 집합들에 연관하는 보조적(auxiliary) 인덱스 데이터 구조가 사용될 수 있다. 여기에서, 만약 보조적 인덱스 데이터를 검색하여 비공 집합이 산출되면(yield), 이 집합을 원하는 후보 집합으로 한다. 아니면 원하는 후보 집합은 입력 검색 패턴을 위해서 (주요) 인덱스 데이터 구조를 검색하는 것에 의해서 획득(obtain)된다.
- [0027] 이러한 방식으로, 본 방법의 효율성이 보다 개선될 수 있다. 연관된 후보 집합이 결정되는 각각의 검색 패턴을 위해서, 주요 인덱스 데이터 구조를 완전히 검색하는 것보다는, 다른 보조적 인덱스 데이터 구조가 사용되어, 객체 속성값 프레픽스들의 부분 집합을 사전-컴팩트화(pre-compactified)되고 사전-조립된(pre-assembled) 후보 집합들로 연관할 수 있다. 전술한 바와 같이, 공간적 원소들을 다음-레벨 공간적 원소들의 감소된 수로 조합하는 것에 의해서, 여기에서 사전-컴팩트화가 수행되나, 보조적 인덱스 데이터 구조가 생성되는 시간에, 즉 검색 쿼리가 처리되는 전에, 일어난다. 만약 보조적 인덱스 데이터 구조 내에서의 검색에서 비공 집합이 산출되면, 원하는 후보 집합이 완성된다. 아니면, 원하는 후보 집합은 (주요) 인덱스 데이터 구조 내에서 검색을 하는 것에 의해서 연산된다. 이 방법은 예컨대 검색 스트링들이 단지 하나의 문자를 함유하는 것과 같은 매우 비표현적 검색 패턴들에서도 후보 집합들을 결정하는 것을 허용한다. 그러한 경우들에 있어서, 인덱스 데이터 구조 내에서 검색 패턴에 정합하는 많은 프레픽스가 발견되고, 후보 집합들 내에 포함하기 위하여 많은 수의 공간적 원소들의 연관된 컬렉션들을 고려해야 할 필요가 있기 때문에, 주요 인덱스 데이터 구조 내에서의 프레픽스를 위한 검색은 매우 효율적이지 않다. 제안된 수정예(modification)에서, 많은 수의 컬렉션들이, 보조적 인덱스 데이터

구조에 의해서 제공되는 보다 작은 수의 제한된-크기의 컬렉션들로 대체된다.

- [0028] 내비게이션 장치 내에서 사용되는 데이터베이스는, 지리적 맵의 디지털 데이터를 포함할 수 있다. 이 경우에서, 객체들은 맵 데이터의 지리적 객체들, 예컨대 거리들 또는 관심 지점들에 대응하고, 2차원 평면은 내비게이션 장치에 의해서 디스플레이될 지리적 맵에 대응한다. 인덱스 데이터 구조는, 이후, 그 안에서 명칭을 지리적 객체들의 속성을 묘사하는 것인 타일들의 근사(approximate) 집합들에 연관한다. 타일들은, 예컨대 대륙 지도와 같은 전체적인 맵의 계층적 쿼드트리 분할로부터 얻어지며, 특정한 이진 순번 스킴[binaray numbering scheme]을 사용하여 번호가 매겨진다. 인덱스 데이터 구조 내에서 일어나는 검색은 하나 또는 다중의 입력 검색 패턴들에 의해서 질의된다(queried). 이로부터, 타일들의 집합은 그 내에서 검색 패턴들이 속성으로서 존재하는 객체들을 포함하는 결과가 된다. 연결적인(conjunctive) 검색, 즉 다중의 입력 검색 패턴들의 조합된 쿼리(논리적 AND-연접)에 대해서, 단일 입력 검색 패턴들의 후보 집합들의 교집합이 형성된다. 결과적인 조합 후보 집합은 이후 모든 입력 검색 패턴들을 속성들로서 포함하는 객체들에 대해서 제2 스텝 내에서 철저하게 검색된다. 이러한 개념은 공통된 프레픽스(단어의 시작)를 포함하는 모든 명칭들에 대해서 후보 집합들을 용이하게 감소시키는 것을 허용한다. 따라서, 효율적인 다중-프레픽스 검색이 가능해진다. 프레픽스와 연관되는 후보 집합을 획득하기 위해서, 프레픽스의 모든 가능한 연속성들(continuations)의 후보 집합들의 조합 집합이 형성된다. 실질적으로[in substance] 연관된 후보 집합들의 지속성(persistent) 묘사 및 인덱스 데이터 구조를 사용하는 것에 의해서, 이러한 프로세스가 메인 메모리의 워크셋(work set)에 증분적으로 부가되는 원소들의 순차적 판독(sequential read-in)을 감소시킨다.
- [0029] 모바일 장치들 상에서 큰 후보 집합들에 대해서도 본 발명을 수행하는 것을 가능하게 하기 위해서, 연산적으로 효율적인 방식 내에서 집합들에 조작들을 적용(apply)하기 위해서 메모리-효율적인 방식 내에서 타일들의 (큰) 집합들을 묘사하는 것이 필요하다. 본원에서의 특정한 묘사는 타일들에 대한 순번 스킴에 기반하고 또한 작은 타일들로부터의 미리 지정된 사각(quadratic) 패턴들이 동등하게 보다 큰 다음-레벨 쿼드트리 타일들로 대체될 수 있다는 것에 기반한다. 타일들을 조합하는 것에 의해서, 저장될 타일들의 수는 감소하고, 묘사는 "컴팩트화"된다.
- [0030] 동등한 컴팩트화(그 내에서 각각의 경우 4개의 타일들이 연관된 다음-레벨 타일과 조합됨) 및 안전 근사 컴팩트화[secure approximate compactification](그 내에서 각각의 경우 4개 또는 그 이하의 타일들이 연관된 다음-레벨 타일과 조합되고, 다음-레벨 타일에 의해서 묘사되는 영역은 원래 타일들에 의해서 커버되는 영역의 상위 집합임) 모두가 사용될 수 있다. 안전 근사 컴팩트화는, 검색을 위해서, 후보 집합 내에 포함된 타일들의 수의 증가를 가져오는 것을 확실하게 하고, 결국 가능하게는 어떠한 실체적(real) 후보 객체들도 그 내에서 위치 배치되지 않는 추가적인 타일들 내가 철저하게 검색되지만 동시에 진실적(true) 후보들을 배제하는(leave out) 묘사 결과의 부정확성을 배제하는 것을 의미한다.
- [0031] 본 발명의 바람직한 실시예에서, 제1 입력 검색 패턴 및/또는 제2 입력 검색 패턴을 상호 작용적으로[interactively] 입력할 때, 조합 후보 집합은 증분적으로 감소된다. 이러한 것의 근거는, 만약 조합 후보 집합이 다중-프레픽스(multi-prefix) 검색 쿼리(하나 또는 다중 입력 검색 패턴들을 포함함)에 대해서 연산되면 그 리고 사용자에게 의해서 하나 또는 다중의 신규(new) 검색 패턴들이 부가되거나 하나 또는 다중 검색 패턴들의 끝 부분에서 하나 이상의 문자들이 추가되거나 하여 검색 쿼리가 변화(alter)되면, (직전에 서술한 바와 같은 방식으로) 확장된 프레픽스에 연관된 후보 집합은 항상 원래 프레픽스에 연관된 후보 집합의 부분 집합이라는 것과, 집합 이론 법칙을 활용하는 것에 의해서, 오래된 조합 후보 집합으로부터 신규 조합 후보 집합이 효율적으로 결정된다. 각각의 변화된 또는 추가된 검색 패턴에 대해서 단지 단일 교집합 조작이 요구된다. 이러한 교집합 조작에서, 신규 또는 변화된 검색 패턴으로부터 획득되는 신규 후보 집합은 오래된 조합 후보 집합과 교차한다. 각각의 경우에서의 조합 후보 집합의 카디널리티는 작아지거나(decrease) 또는 적어도 증가하지는 않는다.
- [0032] 추가적인 실시예에서, 제1 입력 검색 패턴 및/또는 제2 입력 검색 패턴을 상호 작용적으로 입력할 때, 후보 집합의 표현성(expressiveness)에 대한 피드백이 제공된다. 여기에서 후보 집합의 카디널리티가 작아지면, 표현성은 증가한다. 예컨대 후보 집합의 카디널리티가 미리 결정된 임계값보다 작으면 피드백이 제공된다.
- [0033] 피드백은 디스플레이를 통하여 시각적인 방식으로 제공될 수 있지만, 대용량 스피커를 통하여 청각적으로 또는 (예컨대 진동을 발생시키는) 감각 출력 장치를 통하여 감각적으로 제공될 수도 있다.
- [0034] 검색 질의의 입력 또는 저장된 검색 질의의 변화 이후에, 검색 구역을 한정하기 위한 검색 질의의 가능성인 것인 검색 질의(하나 또는 다중 입력 검색 패턴들로 구성됨)의 표현성의 평가(evaluation) 및 신속한 예측을 위해서, 인덱스 데이터 구조의 특정 구조가 허용된다. 결과적인 조합 후보 집합이 작음[그러나 양(positive)인] 카

디널리티를 가진다면, 본 문맥에서는, 검색 질의는 높은 표현성을 가진다.

- [0035] 인덱스 데이터 구조의 특정 지속성 구조는, 집합 내에 포함된 공간적 원소들의 완전한 계수(count)를 수행하지 않더라도 양호한 정확도로 조합 후보 집합의 카디널리티를 예측하는 것을 가능하게 한다. 이에 의해서, 검색 질의가 입력되거나 변화될 때, 제한된 자원을 가지는 모바일 내비게이션 장치도 사용자에게 검색 질의의 실제 표현성에 대한 즉각적인 피드백을 제공할 수 있다.
- [0036] 예컨대, 내비게이션 장치의 디스플레이는, 검색 질의가 적절한 연산적 노력[computational effort]을 가지고 철저하게 검색될 수 있는 조합 후보 집합을 획득하기에 충분한 표현성을 가진다면, 표시되는 입력 윈도우의 붉은 색 배경 착색(colouring)을 노란 색으로 변경하도록 설정될 수 있다. 이 경우에서, 만약 사용자가 입력을 계속하지 않으면, 조합 후보 집합의 공간적 원소들 내에서 발견되는 결과적 객체들을 포함하는 결과적 리스트가 (가능하게는 공간적 원소들의 비교적 큰 수 내에서의 철저한 검색을 위한 대기 시간이 지난 후에) 출력된다. 일반적으로, 원하는 목적지 객체가 결과적 리스트 내에서 포함되는 개연성(probability)은, 검색 질의의 제한된 표현성 때문에 제한되는 편이다. 그러나, 만약 사용자가 입력을 계속한다면, 변화된 검색 질의와 연관된 조합 후보 집합의 일단(once) 예측된 카디널리티가 실질적으로 더 작은 사전-정의된 임계값보다 작아지게 되면, 착색은 녹색으로 변경된다. 이 경우에서, 작은 카디널리티를 가지는 조합 후보 집합이 획득되면, 검색 질의의 높은 표현성 때문에, 발견된 객체들의 획득된 결과적 리스트 내에서 원하는 목적지 객체가 존재할 기회가 많으므로, 조합 후보 집합에 연관된 공간적 원소들에 대한 철저한 검색이 빠르고 효율적인 방식으로 수행될 수 있다.
- [0037] 표현성 예측은 또한 임의의 타일들과도 연관되지 않는 입력의 순간적인 식별[instantaneous identification]을 허용한다. 이 경우에서, 조합 후보 집합의 카디널리티는 0의 값으로 가정한다. 0의 카디널리티는 주로 입력 검색 패턴들의 타이핑 오류들에 의해서, 어떠한 정합하는 객체 속성들도 인덱스 데이터 구조 내에서 발견되지 않으며 비어 있는 후보 집합이 도출되는 것에 기인한다. 그러한 타이핑 오류들은, 따라서, 예컨대 [예컨대 입력 윈도우가 빨간 색으로 점멸(blinking)하는 것과 같은] 시각적, 청각적 또는 감각적 알람 등의 적절한 피드백에 의해서 즉시 지시될 수 있다,
- [0038] 효율적인 검색을 위해서, 공간적 원소들의 조합 후보 집합 내에서 결과적인 객체들에 대해 철저하게 검색하는 것은, 조합 후보 집합의 카디널리티가 미리 결정된 임계값보다 작아진 이후에 수행된다. 철저한 검색 도중 또는 그 이후에, 결과적 객체들이 예컨대 내비게이션 장치의 디스플레이와 같은 출력 장치를 통하여 출력된다.
- [0039] 공간적 원소들의 조합 후보 집합 내에서의 객체들을 철저하게 검색하는 것에 대한 연산적 노력(그리고 사용자에 의해 인지되는 연관된 경과 시간)을 제한하기 위해서, 다음 방안들 중 하나 또는 다수를 취할 수 있다.
- [0040] - 이미 검색된 것인 공간적 원소들의 수가 계수되고, 사전-정의된 한계를 넘어설 때 프로세스가 중단(stop)되거나,
- [0041] - 이미 발견된(found) 객체들의 수가 계수되고, 사전-정의된 한계를 넘어설 때 프로세스가 중단되거나,
- [0042] - 검색에 대한 경과 시간[elapsed time]이 측정되고, 사전-정의된 한계를 넘어설 때 프로세스가 중단되는 것
- [0043] 여기에서, 검색 패턴을 추가하거나 변화하는 각각의 사용자 입력 후에 그리고 신규 또는 변경된 검색 질의로부터 얻어지는 신규 조합 후보 집합을 연산한 후에, 조합 후보 집합 내에서의 객체들에 대한 철저한 검색은 시작되거나 재-시작될 수 있다. 본 문맥에서, 객체들에 대한 철저한 검색은, 보다 높은(보다 거친) 레벨로부터 시작하여 하나 이상의 세밀한(finer) 레벨들로 진행하는 것에 의해서, 인덱스 데이터 구조의 대응되는 레벨들에 대한 두 개 또는 그 이상의 페이즈들(phases) 내에서 수행된다. 제1 페이즈에서는, (쿼드트리 내의) 보다 높은 레벨의 타일들은 철저하게 검색된다. 그러한 타일들은 보다 낮은 레벨들 내의 타일들에 대한 모든 정보를 포함하지는 않고 단지 가장 중요한 정보(예컨대 매우 중요한 관심 지점들이나 대도시들의 명칭들)만을 포함할 수 있다. 모든 정보가 배려(regard)되지 않기 때문에, 제1 페이즈에서의 검색은 따라서 거칠다. 최종 페이즈 내에서, 최하위(lowest) 및 가장 세밀한(finest) 레벨의 타일들이 철저하게 검색되고 그 내에 포함된 모든 정보가 고려될 때까지, 다음 페이즈에서는, 보다 낮은 레벨의 타일들이 이후 검색된다. 이러한 방식으로, 철저한 검색은 점진적으로 보다 정확해지고, 초기 페이즈들 내에서도 이미 결과적 객체들의 리스트가 사용자에게 출력될 수도 있다. 또한 만약 예컨대 시간 한계에 도달하여, 완료 전에 철저한 검색이 중단되면, 적어도 가장 중요한 검색 결과들이 해당 시간에서 이미 디스플레이된다는 것을 보장한다.
- [0044] 다른 실시예에서는, 주요 인덱스 데이터 구조에 부가하여 다른 별도의 인덱스 데이터 구조가 사용될 수 있다. 이는 객체 속성값들을 공간적 원소들의 컬렉션들에 연관하나, 단지 객체가 위치해 있는[situated] 구역을 기술하는 객체 속성들(예컨대 "주변 도시의 명칭", "객체의 우편 번호", "주변 국가의 명칭" 등)에 대해서만 연관하

고 다른 모든 속성들은 무시(disregard)한다. 이러한 별도의 인덱스 데이터 구조는, 따라서 크기 면에서 제한되고, 주요 인덱스 데이터 구조를 사용하여 원하는 구역 내에서 검색하기 전에, 원하는 구역(예컨대 도시)을 신속하게 식별하기 위해서 사용될 수 있다. 별도의 인덱스 데이터 구조를 사용하여 구역에 대해서 우선 검색하는 것에 의해서, 객체들을 위해서 주요 인덱스 데이터 구조를 검색할 때 고려되어야만 할 공간적 원소들의 수가 특정 구역에 연관된 해당 공간적 원소들로 사전에 제한될 수 있고, 따라서 본 방법에 따른 연산적 노력을 더 감소시킬 수 있다.

[0045] 또한 특정한 목표 구역[target region]에 연관되는 공간적 원소들만을 고려하여 검색이 제약될 수도 있다. 만약 (예컨대 사용자의 현재 GPS 위치 주위의 구역과 같은) 특정한 구역으로부터의 검색 결과만 관심이 있다면, 검색 방법은 전술한 바와 같이 적용될 수 있지만, 실제 정합들에 대한 철저한 검색을 수행하기 전에, 추가적으로 후보 집합이 "목표 구역 집합"이라고 지칭되는 공간적 원소들의 다른 집합과 교차할 수 있다. 목표 구역을 기술하거나 이를 상위 집합에 의해서 근사하는(approximating) 공간적 원소들의 집합으로서, 목표 구역 집합은 이러한 목적으로 구성된다. 이 집합이 일반적으로 이미 작은 크기를 가지기 때문에, 이는 초기 후보 집합으로서 사용되어, 입력 검색 패턴들에 따라서 추가적인 후보 집합들을 충분히 연산할 때, 모든 추가적인 교집합들에 적용될 수 있다. 만약 목표 구역 집합이 매우 거친 근사를 사용한다면, 실제 목표 구역 밖의 결과적 객체들은 철저한 검색 도중에 필터링되어 무시될 수도 있다.

[0046] 추가적인 실시예에서, 조합 후보 집합의 철저한 검색 이후에 그리고 모든 입력 검색 패턴들에 정합하는 결과적 객체들의 집합을 획득한 이후에, 결과적 객체들은 타당성이 감소하는 순으로 표시될 수 있다. 결과적 객체들의 집합의 객체들이 사용자가 실제로 발견하기를 의도한 객체(예컨대 거리 주소들이나 관심 지점 등과 같은 목적지)를 나타내는 개연성을 지시하는 휴리스틱(heuristic) 규칙들의 집합에 따라서 객체들의 타당성이 예컨대 결정될 수 있다.

[0047] 검색 결과들의 리스트를 디스플레이할 때, 예컨대 모바일 내비게이션 장치들의 제한된 디스플레이 크기 때문에, 사용자를 위해서, 진실적 목적지 객체가 높은 개연성으로 리스트의 최상부(top) 또는 적어도 최상부 근처에 표시되도록, 그들의 예측된 타당성의 순서에 따라서 제시하는 것이 매우 중요하다. 이를 위해서, 규칙들의 집합이 개발되어서, 개연적(probable) 타당성을 높은 정확도로 평가하는 것을 허용한다. 이러한 규칙들은 한편으로는 사용자에게 의해서 입력된 검색 쿼리의 상세 사항으로부터 유발(arise)하며 하나 또는 다중의 검색 패턴들로 구성된 정보를 사용한다. 다른 한편으로는, 특정한 객체들(예컨대 특정 중요도의 관심 지점들)을 위한 증가된 포괄(generic) 타당성을 나타내며 맵 내에서의 객체 속성들로 저장되는 정보를 사용한다.

[0048] 휴리스틱 규칙들은, 예컨대

[0049] - (예컨대 제1 검색 패턴은 도시를 지시하고 제2 검색 패턴은 거리를 지시하는) 사용자가 선택한 입력 검색 패턴들의 순서를 고려하는 것.

[0050] - 구역들/도시들을 거리들보다 더 중요하게 고려하는 것.

[0051] - 대도시들을 소도시들보다 더 중요하게 고려하는 것.

[0052] - 공항들 또는 도시들에 대응하는 관심 지점들을 다른 관심 지점들보다 더 중요하게 고려하는 것.

[0053] - 관심 지점들을 거리들보다 더 중요하게 고려하는 것 및/또는.

[0054] - 큰 거리들을 작은 거리들보다 더 중요하게 고려하는 것이다.

[0055] 내비게이션 장치 내에서, 데이터베이스는 지리적(geographic) 객체들에 대응하는 맵 데이터를 포함한다. 추가적인 실시예에서, 제1 입력 검색 패턴 및/또는 제2 입력 검색 패턴을 상호 작용적으로 입력할 때, 공간적 원소들의 조합 집합에 대응하는 지리적 영역이 출력된다. 이 경우 각각의 검색 쿼리를 위해서, 맵 타일들의 조합 후보 집합이 연산되고, 각각의 그러한 집합들에 대해서, 그에 대응하는 지리적 영역 또는 개별적 조합 후보 집합을 포함하는 지리적 영역이 결정된다. 이 영역은 예컨대 조합 후보 집합 내에 포함된 모든 타일들의 최북단, 최남단, 최서단(westernmost), 최동단(easternmost) 에지들에 의해서 각 변들이 결정되는 직사각형일 수 있다. 검색 쿼리의 입력 후에 이 직사각형을 묘사하는 맵 세그먼트를 즉각(prompt) 디스플레이하는 것으로, 잠재적으로(potentially) 원하는 목적지가 발견될 수 있는 영역의 신속한 오버뷰(overview)를 제공할 수 있다.

[0056] 바람직하게는, 여기에서, 지리적 영역의 디스플레이는, 일반적으로 연산 시간을 소모하는 것인 조합 후보 집합 내에서 객체들의 철저한 검색을 수행하기 전에 갱신될 수 있다.

- [0057] 제1 입력 검색 패턴 및/또는 제2 입력 검색 패턴을 상호 작용적으로 입력할 때, 공간적 원소들의 조합 집합에 대응하는 지리적 영역이 증분적으로 국한될 수 있다. 이러한 방식으로, 검색 쿼리의 입력 동안에 후보 집합의 효율적인 증분적 국한과의 조합 내에서, (효율적인 맵 시각화가 가능하다는 전체에서) 특정한 시각적 인상(impression)이 획득될 수 있다. 그 내에서, 검색 쿼리의 점진적(progressive)이고 증분적인 입력에, 잠재적인 원하는 목적지가 안에 위치 배치되는 영역을 묘사하는 지리적 영역을 연속적으로 주밍하는 것이 동반(accompany)된다. 지리적 영역은, 여기에서는, 전체 대륙을 도시하는 맵으로부터 목적지 객체(또는 다른 객체들) 및 그의 가까운 환경만을 포함하는 작은 맵 세그먼트까지의 범위의 맵 세그먼트에 대응할 수 있다. 충분한 디스플레이 속도와 (가능하게는 인터플레이션 기술들을 이용한) 디스플레이 스케일의 연속적 조정(adjustment)으로, 영화 카메라의 줌 효과와 유사한 시각적 인상이 획득될 수 있다. 맵 디스플레이의 속도를 개선하기 위해서, 예컨대 숲 영역과 같은 중요도가 작은 지도 상세 사항들을 배제하거나 맵 내에서 표시되는 텍스트를 감소시키는 것에 의해서, 디스플레이된 맵 세그먼트를 실질적으로 단순화하는 것이 가능하다.
- [0058] 추가적인 실시예에서, 제1 입력 검색 패턴 및/또는 제2 입력 검색 패턴은, 특히 복수의 문자들 및/또는 숫자들이 키패드의 하나의 키에 할당되어 있는 간략화 키패드(reduced keypad)를 통하여 입력되고, 각각의 키는 입력 검색 패턴의 문자 당 단 한번만 눌러진다. 어느 정도로 관련되지 않는[uncorrelated] 자연어의 단어들 또는 지리적 명칭의 인근(adjacent) 문자들에서조차도, 명확한 통계적 특성을 이용하여, 이러한 방식으로, 결과적인 불충분 결정[underdetermination]은 일반적으로 모든 추가적인 입력 문자들에 의해서 감소될 수 있다.
- [0059] 제1 변종예(variant)에서는, 단계들의 복잡한 시퀀스를 사용하여 모바일 장치들 상의 가상 또는 물리적 키패드들을 통하여 종래 입력되는 것인 구별적(diacritical) 문자들[움라우트 기호들(umlauts)]의 입력은 회피(avoid)될 수 있다. 이 실시예에서는, 기본 문자 및 움라우트가 정합되는 것으로 고려되는 검색 프로세스 내에서, 검색 패턴의 일부로서, 움라우트에 연관된 기본 문자(예컨대 "Ä" 대신에 "A")를 입력하는 것이 가능해진다.
- [0060] 제2 변종예에서는, 다중의 알파벳 문자들이 각각의 숫자 키에 할당(assign)된 것("0" 내지 "9", "*", "#") 키들이 통상적으로 공지되듯이 "2"="ABC", "3"="DEF"으로 할당된 것)인, 이동 전화(내비게이션 애플리케이션을 포함하고, 내비게이션 장치로서 기능함)의 숫자 키패드가 사용될 수 있다. 검색 패턴의 입력을 위해서, 어떠한 다중 키 스트로크들도 요구되지 않는다(예컨대 "C"를 위해서, "2"번 키는 단지 한번만 눌러진다). 사용자는 이러한 방식으로 마치 완전한 전체 키보드를 사용하는 것처럼 빠른 방식으로 검색 쿼리를 입력할 수 있다. 타이핑 오류들은 따라서, 예컨대 터치스크린 상에 표시되는 것과 같은 축소화된 전체 키보드에 비해서 개연성이 작다. 이러한 개념을 사용한 방법을 실질적으로 구현하는 것은 더 복잡해지는 것이 아니고, 본 방법의 실행은, 전체 키보드를 통하여 입력을 하는 것에 비해서 연산 능력 면에서 단지 조금 비용이 들어가는 것 뿐이다.
- [0061] 장치 내에서, 디스플레이 상에 디스플레이되는 가상적 및 물리적 키보드가 사용될 수 있다. 가상적 키보드 내에서, 가상 키들은, 포인팅(pointing) 장치, 방향성(directional) 키패드, 특정 트리거 키들, 또는 터치-민감 스크린을 사용하여 작동(actuate)될 수 있다.
- [0062] 본 방법의 실시예는, 특정한 방식 내에서 검색 쿼리의 입력 검색 패턴을 해석하는 사상에 기초하는 것인 다중-페이지 휴리스틱 검색 전략을 사용할 수 있다. 이 문맥 내에서, 특정한 해석을 위해서, 소위 역할(role)이 사용자에 의해서 입력되는 검색 쿼리의 각각의 입력 검색 패턴에 대해서 할당된다.
- [0063] 근원적인(underlying) 사상은 다음 관측으로부터 기원한다: (예컨대 지리적 맵 내인) 데이터베이스 내의 특정 목표 객체를 발견하려는 의도로 사용자가 특정 검색 쿼리를 입력할 때, 이는 이 검색 쿼리에 기초하여 고유의(uniquely) 목표 객체를 가능한 식별하는 것을 사용자가 의도하는 것을 의미하고, 이는 보통(항상은 아니지만) 실제적인 경우이다.
- [0064] 이제, 비록 사용자가 이러한 사항을 상세하고 명쾌하게(explicitly) 생각하지는 않더라도, 사용자는 만약 질문을 받으면, 목표 객체를 명확하게 식별하기 위한 방식으로 어떻게 이러한 검색 쿼리가 이해될지를 보통 설명할 수 있다. 따라서 본 방법은 "단지" 이러한 해석을 "추측(guess)"하는 것만이 필요하다.
- [0065] 이러한 것은 특정한 방식으로 검색 쿼리의 입력 검색 패턴 각각의 역할을 할당하는 것에 의해서 입력 검색 쿼리를 해석하는 것으로 수행될 수 있다.
- [0066] 특정한 역할의 가능한 예는 다음과 같다: "객체의 구역 속성 내에서 발생하는 검색 패턴" 또는 "검색 패턴은 객체의 명칭의 첫 번째 단어이어야 한다". 따라서 본 문맥에서 "역할"이라는 용어는, 사용자가 의도하고 있는 입력 검색 패턴의 가능한 특성을 지칭하는 것으로 이해하여야 한다. 예컨대 사용자가 검색 쿼리 "BER HUS"를 입력

한다면, 제1 입력 검색 패턴("BER")은 객체의 구역 속성(예컨대 베를린 시)을 지시하는 것으로 해석될 수 있고, 제1 입력 검색 패턴("HUS")은 객체의 명칭(예컨대 베를린 시 내의 거리인 객체 "Husemannstraße")을 지칭하는 것으로 해석될 수 있다. 다른 해석들도 물론 가능하다.

[0067] 검색 패턴들에 대해서 역할들을 할당하는 것에 의해서 검색 쿼리를 해석하는 것은 검색 공간을 제한하는 것에 사용될 수 있다. 즉, 검색 쿼리의 특정한 해석을 위해서, 가능하게는, 예컨대 분할된 맵 데이터에 의해서 정의되는 데이터베이스의 전체 검색 공간이 아니라, 단지 제한된 검색 공간만을 검색하는 것이 필요하다. 위의 예시에서, 만약 "BER"이 구역 속성으로서 해석된다면, 단지 구역 속성들만을 고려한 (제한된 보조적) 인덱스 데이터 구조가, 구역에 관계된 객체의 속성들(즉 구역 속성)만을 검색하기 위해서 사용될 수 있다.

[0068] 본 방법의 실시예에서는, 따라서, 제1 입력 검색 패턴 및/또는 제2 입력 검색 패턴은, 특정한 사전-정의된 역할들을 가지도록 휴리스틱하게 해석되고, 여기에서 역할은 개별적인 입력 검색 패턴의 가능한 근원적인 의미를 지시한다.

[0069] 제1 입력 검색 패턴 및/또는 제2 입력 검색 패턴을 휴리스틱하게 해석하는 것에 의해서, 역할-특화(role-specific) 휴리스틱 후보 집합이 각각의 입력 검색 패턴에 대해서 연산되고, 역할-특화 휴리스틱 후보 집합을 해석하는 것에 의해서, 검색 쿼리의 특정한 해석을 위해서 조합 휴리스틱 후보 집합이 획득된다.

[0070] 본 방법 내에서, 각각의 경우에서의 검색 쿼리의 여러 개의 서로 다른 해석을 위한 제1 스텝에서는, 조합 휴리스틱 후보 집합이 연산되고 철저하게 검색되고, 제2 스텝에서는, 검색 쿼리를 특정한 해석에 제약(constraining)하지 않는 조합 후보 집합이 연산되고 철저하게 검색된다.

[0071] 검색은 따라서 서로 다른 페이즈들로 분리된다. 첫째로, 검색 쿼리의 몇 가지의 가능한 해석들이 고려되어, 원하는 결과를 포함하는 양호한 기회를 가지며 각각이 신속하고 철저하게 검색될 수 있는 검색 공간의 매우 작은 프랙션(fraction)이 각각 산출된다. 둘째로, 본 방법은 보다 큰 검색 공간을 산출하지만 데이터베이스 내에 존재한다면 원하는 결과를 포함하는 것이 보증되는 일반적인 해석으로 돌아간다. 만약 철저한 검색 도중에 자원 한계(resource limit)에 도달하면, 검색 엔진은 검색 프로세스를 중지(abort)하고 애플리케이션에 통지(notify)하여, 보다 표현성인 입력을 제공하도록 사용자에게 권장(encourage)한다.

[0072] 전술하듯이, 해석은 입력 검색 패턴들 각각에 특정한 역할을 할당한다. 입력 검색 쿼리의 서로 다른 해석들을 산출하는 서로 다른 역할들의 가능한 조합들이 많기 때문에, 연관된 역할-특화 휴리스틱 후보 집합들 및 조합 휴리스틱 후보 집합들을 연산하지 않으면서 서로 다른 해석들이 우선 상징적으로(symbolically) 열거(enumerate)된다. 이후, 하나 이상의 역할-특화 휴리스틱 후보 집합들이 인덱스 데이터 구조 내에서 룩업(lookup)을 통하여 비어 있는 것으로 검출(detected)된 것에 대해서, 이러한 해석들이 더 이상 고려되지 않는다. 다른 말로 하면, 본 방법의 실시예에서는, 실제 후보 집합들을 연산하기 전에, 모든 비공 해석들은 열거되고 정리되고(ordered), 따라서 보다 작은 후보 집합들을 산출하는(그리고 가능하게는 원하는 결과를 산출하는) 것이 기대되는 해석들이 우선 나온다. 본 문맥에서, 역할이 지정된 검색 쿼리의 임의의 검색 패턴이 비어 있는 역할-특화 휴리스틱 후보 집합을 산출한다면, 주어진 검색 쿼리에 대해서 해석은 "비어 있는" 것이라고 지칭된다. 이것은 자동적으로 해석의 조합 휴리스틱 후보 집합을 비어 있는 것으로 하게 되고, 따라서 그러한 해석들은 더 이상 고려될 필요가 없다.

[0073] 이러한 점에서, 입력 검색 패턴을 위해서 고려되는 서로 다른 해석들의 선택 또는 서로 다른 해석들의 상징적 열거는, 인덱스 데이터 구조 내에서 룩업을 통하여 획득되는 하나 이상의 역할-특화 휴리스틱 후보 집합들의 예측된(estimated) 카디널리티에 따라서 결정될 수 있다.

[0074] 검색 쿼리의 입력 검색 패턴에 특정한 역할을 할당하는 사상과, 특정한 역할을 가지고 검색 패턴들을 포함하는 검색 쿼리에 따라서 데이터베이스를 검색하는 것에 의해서 검색의 효율성을 개선하는 것은 또한 분리하여 적용될 수 있다. 따라서, 본 방법은 예컨대 쿼드트리의 타일들, 옥트트리의 직육면체들, 이진 트리의 노드들, 문서들, 파일들, 또는 특정 문서들 또는 파일들을 식별하기 위한 숫자들 또는 다른 코드들, 또는 문서, 파일 또는 공간적 원소 내에서의 속성값의 하나 이상의 발생 위치(occurrence position)들 및 전술한 정보 중 임의의 것을 포함하는 데이터 구조들인 것인 원소들의 컬렉션들에 데이터베이스 객체 속성값들을 연관하는 인덱스 데이터 구조에 의해서 데이터베이스 내의 객체들을 검색하는 방법으로서,

[0075] 입력 검색 쿼리 내에서, 휴리스틱하게 제1 입력 검색 패턴 및/또는 제2 입력 검색 패턴을 특정한 사전-정의된 역할들을 가지도록 해석하는 단계로서, 여기서 역할은 객체 속성값에 대한 개별적인 입력 검색 패턴의 가능한

근원적인 의미를 지시하는 것인 단계;

- [0076] 휴리스틱하게 제1 입력 검색 패턴 및/또는 제2 입력 검색 패턴을 해석하는 것에 의해서, 각각의 입력 검색 패턴을 위해서 인덱스 데이터 구조에 의해서 원소들의 역할-특화 휴리스틱 후보 집합을 연산하는 단계로서, 역할-특화 휴리스틱 후보 집합은 미리-기술된(pre-scribed) 역할 내에서 입력 검색 패턴을 정합하는 적어도 하나의 속성값을 포함하는 원소들의 컬렉션들을 포함하는 것인 단계;
- [0077] 검색 쿼리의 특정한 해석을 위해서 조합 휴리스틱 후보 집합을 획득하도록 제1 입력 검색 패턴 및 제2 입력 검색 패턴을 위해서 역할-특화 휴리스틱 후보 집합을 교차하는 단계; 및
- [0078] 원소들의 조합 휴리스틱 후보 집합 내에서, 제1 입력 검색 패턴 및 제2 입력 검색 패턴에 정합하는 객체들을 식별하여, 결과적 객체들의 집합을 획득하는 단계
- [0079] 를 포함한다.
- [0080] 본 방법의 실시예에서, 각각의 경우에서의 검색 쿼리의 여러 개의 서로 다른 해석을 위한 제1 스텝에서는, 조합 휴리스틱 후보 집합이 연산되고 철저하게 검색되고, 제2 스텝에서는, 검색 쿼리를 특정한 해석에 제약하지 않는 조합 후보 집합이 연산되고 철저하게 검색된다.
- [0081] 전술한 본 방법의 배경 및 장점에 대해서 서술한다.
- [0082] 일반적으로, 본 방법의 서로 다른 실시예에 대해서, 주요 인덱스 데이터 구조가 적용되기 전에 보조적 인덱스 데이터 구조는 항상 검사(probe)되고, 특정한 입력 검색 패턴들을 위한 사전-컴팩트화된 후보 집합들을 제공하는 보조적 인덱스 데이터 구조를 적용하여, 낮은 표현성을 가지는 입력 검색 패턴들에 대한 후보 집합들을 연산하기 위한 연산 노력은 감소될 수 있다. 즉 특정한 매우 비표현적 검색 패턴들(예컨대 단일 문자들 또는 단지 2개의 문자들을 가지는 검색 패턴들)에 대해서, 첫째로 그 내에서 비표현적 검색 패턴들이 입력 검색 패턴에 대응하는 원하는 객체를 포함할 수 있는 공간적 원소들의 사전-정의된 컬렉션들과 연관되는 것인, 보조적 데이터가 특업된다. 이러한 것의 배경은, 만약 사용자가 의도적으로(deliberately) 짧고 비표현적인 검색 패턴을 입력한다면, 사용자는 이 검색 패턴을 원하는 객체를 실제로 지향(point toward)하는 것으로 고려하는 가능성이 높고, 예컨대 검색 패턴이 정확하게 객체의 명칭 또는 구역 속성 내에 포함되기 때문이다(예컨대, "P-Street"라고 지칭되는 거리가 있다면, 사용자가 "P"를 입력한다면, 이것을 찾는 것일 수 있다). 만약 이것이 예컨대 공간적 원소들이 (비표현적) 입력 검색 패턴에 정확하게 정합하는 속성을 가지는 객체들을 포함하는 보조적 인덱스 데이터 구조 내에서 사전-저장되어(pre-stored) 있고 (주요) 인덱스 데이터 구조에 앞서서 보조적 인덱스 데이터 구조가 특업된다면, 공간적 원소들의 결과적 후보 집합을 철저하게 검색하는 것에 의해서, 공간적 원소들의 많은 수를 철저하게 검색하지 않더라도 원하는 결과가 신속하게 발견될 수 있다.
- [0083] 본 발명의 추가적인 실시예에서, 주어진 입력 검색 패턴에 대해서 우선 연산된 후에, 후보 집합들 및 조합 후보 집합들(또한 전술한 예컨대 역할-특화 휴리스틱 후보 집합들 및 조합 휴리스틱 후보 집합들을 포함함)은 캐시(cache) 메모리 데이터베이스 내에 저장될 수 있다. 이러한 방식으로, 검색의 서로 다른 스텝들 도중에 여러 번 필요로 하더라도 후보 집합들이 다시 연산될 필요가 없이 단지 캐시 메모리로부터 획득될 수 있으므로, 입력 검색 패턴들에 대한 후보 집합들을 연산하기 위한 연산 노력은 감소될 수 있다.
- [0084] 본 방법의, 추가적으로 개선된 실시예에서는, 인덱스 데이터 구조는, 객체 속성의 정확한 값들 뿐만 아니라 추가적으로 이러한 값들의 사전-정의된 변화량들(variations)을 위하여 엔트리들(entries)을 가지는 것이고, 이는 특정한 문자들을 다른 문자들로 대체하거나, 또는 특정한 문자들을 원래의 값에 부가하거나, 또는 원래의 값에서 특정한 문자를 생략(omit)하거나, 또는 원래의 값 내에서 위치들에 대해서 일부의 문자들의 위치들을 교환(exchange)하는 것이다. 이러한 것은 또한 소위 퍼지(fuzzy) 검색을 위해서 허용되며, 만약 입력 검색 패턴들의 하나 또는 다수가 객체 속성값, 예컨대 객체의 명칭이나 구역 속성에 (정확히) 정합하지 않는 경우에도, 예컨대 원하는 결과를 발견하는 것을 허용한다. 이러한 방식으로, 예컨대 입력 검색 패턴 내에 철자(spelling) 오류가 있는 경우에도, 원하는 결과가 발견될 수 있다.
- [0085] 다른 실시예에서는, 제1 및 제2 입력 검색 패턴을 위한 제1 및/또는 제2 후보 집합은, 공간적 원소들을 제1 및 제2 후보 집합 내로 포함하는 것에 의해서 확장(extend)될 수 있고, 제1 및 제2 후보 집합이 조합되기 전에, 제1 및 제2 후보 집합 내에 이미 존재하는 공간적 원소들에 지리적으로 인접한다. 이러한 방식으로, 결정된 후보 집합의 공간적 원소들에 인접하는 공간적 원소들은 검색 내로 포함된다.
- [0086] 또 다른 본 방법의 변형예에서는, 예컨대 특정한 문자들을 다른 문자들로 대체하거나, 또는 특정한 문자들을 원

래 검색 패턴에 부가하거나, 또는 원래의 값에서 특정한 문자를 생략하거나, 또는 원래 검색 패턴 내에서 그들의 위치들에 대해서 일부의 문자들의 위치들을 교환하는 것으로, 사용자-특화 검색 패턴의 수정예를 하나 이상 생성하는 것에 의해서 소위 퍼지 검색을 지원한다. 이러한 변형예에서는, 본 방법에 따른 인덱스 데이터 구조들은, 이후 서술되는 그러한 검색 패턴 수정예들의 부분 집합을 식별하기 위해서 그리고 선택된 수정예들에 대한 역할-특화 후보 집합들 또는 하나 이상의 후보 집합들을 획득하기 위해서, 사용된다. 이러한 집합들은 곧 개별적으로 사용되거나, 또는 그들의 집합 유니온을 구축한 후, 본 방법에 따른 추가적인 스텝들에서, 조합 후보 집합들의 결정 등에서 사용된다.

[0087] 본 방법의 또 다른 변형예에서는, 제1 및 제2 입력 검색 패턴은 음성 입력 장치 또는 다른 서브시스템(subsystem)과 같은 매체를 통하여 입력되고, 단일 검색 쿼리 대신에 여러 개의 입력 서치 패턴들을 각각 구성하는 다수의 검색 쿼리 변형들(variants)이 산출되고, 검색 쿼리 변형들로부터 검색 패턴을 위해서 연산된 후보 집합들의 카디널리티 또는 검색 쿼리 변형들을 위해 연산된 조합 후보 집합들의 카디널리티가 사용되어, 추가적인 처리를 위한 검색 쿼리 변형들의 부분 집합(subset)의 선택을 유도(guide)하거나 추가적인 처리를 위한 순서에 영향을 미치는 개별적인 검색 쿼리 변형들에 대한 우선 순위들(priorities)을 할당하는 것을 유도한다. 따라서, 예컨대 음성 입력을 사용한 [분명히 규정되지 않은(indefinite)] 입력에 대응할 수 있는 서로 다른 검색 쿼리 변형들을 고려하는 것에 의해서, 예컨대 가장 작은 카디널리티 후보 집합을 산출하는 첫 번째 검색 쿼리 변형을 처리하는 것에 의해서, 특히 이를 변형들로부터 얻어지는 후보 집합들의 카디널리티를 고려할 때, 원하는 결과는 쉽고 효율적인 방식으로 발견될 수 있다.

[0088] 본 방법의 또 다른 변형예에서는, 예컨대 문자들 "0" 및 "Ö" 그리고 숫자 "0"과 같이 특정 입력 문자들이 입력 도중에 신뢰성있게 구별(differentiate)되는 것이 어려운 것인 필기 인식 장치(hand writing recognition device) 또는 서브시스템을 통하여, 검색 패턴들은 입력될 수 있다. 따라서, 본 방법에 따라서 제시된 입력 검색 패턴 내에서 심볼(symbol)의 잘못된 변형이 발생하는 위험성이 존재한다. 본 변형예에서는, 모호성(ambiguity)이 발생할 수 있는 심볼들은 인덱스 데이터 구조가 생성되기 전에 식별될 수 있고, 모호성은 다음과 같이 취급된다: 신뢰성있게 구별될 수 없는 심볼들의 각 집합으로부터, 정통(canonical) 대표라고 지칭되는 고유 대표(unique representative)(예컨대 문자 "0")가 선택된다. 인덱스 생성 도중에 및 철저한 검색 도중에, 심볼의 다른 변형들(예컨대 문자 "Ö" 그리고 숫자 "0")의 모든 발생은 정통 대표로 대체된다. 다른 말로 하면, 숫자 "0"이 또한 문자들 "0" 및 "Ö"로 부과된(overloaded) 간략화 키패드 상에서 발생하는 것과 같이 취급된다.

[0089] 본 방법의 또 다른 변형예에서는, [연산적으로 저가(inexpensive)] 인덱스 데이터 구조 내의 룩업을 통하여 획득되는 것인 하나 이상의 후보 집합들의 예측된 카디널리티는, 입력 검색 패턴들의 다수 또는 입력 검색 패턴의 표현성(expressivity)을 측정하는데 사용된다. 카디널리티를 사용하여, 추가적인 처리 스텝들에서 포함되는 (내부 또는 외부의) 데이터베이스들 또는 데이터베이스 세그먼트들의 선택이, 또는 추가 처리의 순서에 영향을 미치는 것인 그러한 데이터베이스들 또는 데이터베이스 세그먼트들의 우선 순위들의 할당이, 제어될 수 있다.

[0090] 미래의(future) 검색 프로세스들을 위해서 사용되도록, 추가적인 처리(예컨대 내비게이션 애플리케이션 내에서 경로 연산을 위해서 목적지를 선택하는 것)를 위한 검색 결과의 사용자의 최종 선택이 지속성 저장 매체에 등록될 수 있다. 따라서, 이전에 선택된 검색 결과들이 더 큰 우선 순위에 할당되고 나중의 검색의 실제 결과 리스트의 최상부 또는 그 근처에 디스플레이될 수 있는 것인 자체-학습(self-learning) 시스템이 제공될 수 있다.

[0091] 또 다른 실시예에서는, 공간적 원소들의 조합 후보 집합의 철저한 검색의 결과들로서, 입력 검색 쿼리의 모든 입력 검색 패턴들에 정합하는 속성을 가지는 객체들이 등록되는 것 뿐만 아니라, 객체 속성에 정합하지 않는 하나 이상의 입력 검색 패턴들에 대한 객체들도 등록될 수 있다. 따라서, 단어-레벨 퍼지 정합이라고 지칭되는 것이 구현되어, 검색 질의에 정확하게 정합하는 속성들을 가지는 객체들 뿐만 아니라 입력 검색 패턴들의 하나 또는 일부에 대해서 정합을 가지지 않는 객체들도 결과로서 산출된다.

[0092] 또 다른 변형예에서는, 공간적 원소들의 연관된 컬렉션들 및 인덱스 데이터 구조는, 인덱스 데이터 구조를 사용하여 실제로 검색되는 데이터베이스로부터 생성되는 것이 아니고, 실제 검색되는 데이터베이스의 모든 또는 대부분의 객체들을 포함하는 더 큰 데이터베이스로부터 생성될 수 있다.

[0093] 전술한 사항에서, 본 발명에 따른 방법은, 내비게이션 장치에서의 2차원 맵들을 취급할 때 바람직하게 사용되는 것인, 쿼드트리의 2차원 공간적 원소들(타일들)의 컬렉션들에 대해서 객체 속성값들을 연관하는 인덱스 데이터 구조에 초점을 맞추어서 설명되었다. 예컨대 3차원 공간 내에서의 내비게이션 장치(예컨대 항공 또는 우주 애플

리케이션들)과 같은 3차원 공간에서 객체들을 검색하는 방법에도 적용될 수 있다. 이 경우, 인덱스 데이터 구조는, 그 내에서 각각의 공간적 요소가 큐보이드에 대응하고 8개의 인접한 큐보이드들이 다음-레벨 공간적 원소를 묘사하는 하나의 다음-레벨 큐보이드에 조합 가능한 옥트트리(Octree)의 공간적 원소들의 컬렉션들에, 객체 속성값들을 연관한다. 전술한 실시예들과 변형들은 마찬가지로 그러한 인덱스 데이터 구조에 적용될 수 있다.

[0094] 가능하게는, 그보다 높은 차원의 공간들 도는 1차원 공간에도 적용될 수 있다. 일반적으로, 본 방법은, k 는 자연수($k=1, 2, 3, 4, 5, \dots$)인, k -트리의 공간적 요소들의 컬렉션들에 객체 속성값들을 연관하는 인덱스 데이터 구조를 사용할 수도 있다.

[0095] 1차원의 경우, 인덱스 데이터 구조는 1차원 선형(linear) 공간으로부터의 간격(interval)을 묘사하는 각각의 트리 노드인, $k=1$ 인 k -트리로서 해석되는 이진 트리의 원소들을 객체 속성값들에 연관한다. 두 개의 인접한 간격들은, 다음-레벨 트리 노드에 의해서 묘사되는 다음-레벨 간격에 조합 가능하다. 이 경우, 각각의 간격은 1차원 공간을 분할하는 공간적 원소들로서 이해되어야 한다.

[0096] 본 발명의 목적은 전술한 방법을 실행하도록 구성된 내비게이션 장치에 의해서 또는 본 방법을 실행하도록 구성된 다른 임의의 모바일 컴퓨팅 장치에 의해서 달성된다.

발명의 효과

[0097] 본 발명에 따르면 모바일 내비게이션 장치의 맵 데이터 내에서 객체들을 위해서 자유 텍스트 검색이 허용된다.

도면의 간단한 설명

[0098] 후속하여, 본 발명에 따른 기술적 사상들이 도면들에 도시된 실시예들을 참조로 하여 상세하게 설명된다.

도 1은 인덱스 데이터 구조에 의해서 데이터베이스 내에서 객체들을 검색하는 방법의 흐름도이다.

도 2는 트라이를 형성하며 맵을 분할하기 위해서 정의되는 타일들의 컬렉션들에 대해서 객체 속성값들을 연관하는 인덱스 데이터 구조를 도시한다.

도 3a 내지 도 3c는, 내비게이션 장치의 검색 패턴을 입력하기 위한 입력 윈도우와 결과적 객체들의 리스트를 출력하기 위한 출력 윈도우의 도면이다.

도 4a 내지 도 4c는, 검색 패턴을 입력하기 위한 입력 윈도우와, 결과적 객체들의 리스트를 출력하기 위한 출력 윈도우와, 검색 패턴을 입력하기 위한 내비게이션 장치의 키패드의 도면이다.

도 5a 내지 도 5c는, 검색 패턴을 입력하기 위한 제1 및 제2 입력 윈도우와, 결과적 객체들의 리스트를 출력하기 위한 출력 윈도우와, 입력 검색 패턴에 독립적으로 연속적으로 조정되고 디스플레이되는 지리적 맵 세그먼트의 도면이다.

도 6은 얼마간의(a number of) 맵 타일들로 분할된 맵의 도식적 묘사를 나타낸다.

도 7은 휴리스틱 검색 접근을 사용한 변형예의 흐름도이다.

도 8은 다중 데이터베이스들에 걸친 검색 방법의 흐름도이다.

도 9는 사용자가 쿼리를 음성 입력 장치를 통하여 입력하는 문맥으로 본원에서 설명한 방법의 구조적 도시이다.

발명을 실시하기 위한 구체적인 내용

[0099] 도 1은, 흐름도를 통하여, 데이터베이스 내에서 객체들을 발견하고 위치 배치하기 위한 방법의 실시예의 개관을 나타낸다. 본 실시예에서, 본 방법은 내비게이션 장치에 적용되고, 객체들은 2차원 맵의 지리적 객체들을 묘사하며, 데이터베이스 내에 저장된 데이터는 디지털적으로 저장된 지리적 맵 데이터이다.

[0100] 내비게이션 장치는, 예컨대, 전용(dedicated) 모바일 내비게이션 장치, 모바일 컴퓨터, PDA(personal digital assistant), 모바일 인터넷 장치(MID), 휴대용 미디어 재생기(PMP), 또는 다른 모바일 컴퓨팅 장치이고, 그 상에서 내비게이션 소프트웨어 애플리케이션이 구현된다.

[0101] 본 방법에서, 객체들은 소위 트라이라고 지칭되는 것으로 구현되는 인덱스 데이터 구조를 사용하여 발견되고 위치 배치된다. 트라이(200)를 묘사하는 그러한 인덱스 데이터 구조의 개요도가 도 2에 도시된다. 트라이(200)는 지리적 객체들의 명칭을 그 내에서 객체들의 속성들로서 명칭이 발생하는 맵 타일들의 컬렉션에 연관한다. 맵

타일들은, 여기에서는, 공간적 원소들을 묘사하고, 쿼드트리[쿼드트리 분할]을 사용하는 것에 의해서 내비게이션 장치의 구조적 분할로부터 기인한다(arise from). 쿼드트리 분할은, 4개의 인접하는 타일들이 다음-레벨 타일에 조합 가능하도록, 즉 4개의 인접하는 타일들이 임의의 정보도 손실(lost)되지 않으면서 동등하게 다음-레벨 타일에 의해서 대체되는 것이다. 여기에서, 다음-레벨의 영역은 원래의 4개의 인접한 타일들의 영역을 동등하게 커버한다.

[0102] 이러한 것이 도 6에 도시된다. 맵(5)은 얼마간의 맵 타일들(51)로 구조적으로 분할되고, 각각은 맵(5)의 공간적 원소를 묘사한다. 4개의 인접하는 맵 타일들(51)은, 4개의 해당 맵 타일들(51)을 임의의 정보도 손실되지 않고 동등하게 대체하는 것인 다음-레벨 타일(52)에 대해서 조합 가능하다.

[0103] 맵 데이터 내의 객체들을 검색하기 위해서, 검색 쿼리가 내비게이션 장치의 입력 윈도우(1)(도 1)에 입력되며, 검색 쿼리는 텍스트 스트링들 형태의 하나 또는 다수의 검색 패턴들로 구성된다. 도 1에 도시된 경우에서는, "KAP" 및 "WAR"이 검색 패턴들로서 입력된다. 데이터베이스 내의 검색은 이제 각각의 검색 패턴 "KAP" 및 "WAR"에 대해서 개별적으로 인덱스 데이터 구조(트라이 200)를 질의하는 것에 의해서 발생한다(스텝 101). 각각의 이러한 쿼리들로부터, 타일들의 후보 집합(102a, 102b)이 도출되고, 각각의 후보 집합(102a, 102b)은, 그 내에서 적어도 하나의 객체의 속성이 개별적인 검색 스트링 "KAP" 또는 "WAR"와 정합하는 것인 모든 이러한 타일들을 포함한다. 여기에서, 만약 객체의 속성(예컨대 그의 명칭 또는 그의 구역 속성)이 개별적인 검색 패턴 전체 또는 그 프레픽스(즉 그의 시작 문자들)과 정합한다면, 정합은 발생한다.

[0104] 예컨대 모바일 내비게이션 장치들과 같은 제한된 하드웨어 자원들을 가지는 장치들 상에서 전송한 검색 방법을 구현하고 실행하는 것이 가능하도록 하기 위해서, 메모리 효율적인 방식으로 그리고 연산적으로 효율적인 집합 조작들[set operations]을 적용하는 것을 허용하는 방식으로, 타일들의 큰 집합들을 묘사하는 것이 필요하다. 이를 위해서, 4개의 인접하는 타일들이 하나의 다음-레벨 타일에 조합될 수 있어서 임의의 정보도 손실되지 않고 타일들의 수를 감소시킬 수 있는 것인, 쿼드트리의 특정 구조가 사용된다.

[0105] 따라서 획득된 후보 집합(102a, 102b) 내의 타일들의 수가 사전-정의된 수를 초과하면, 스텝 102에서, 후보 집합(102a, 102b) 내에 포함된 일부 또는 전부의 타일들[도 6의 맵 타일들(51)]을 연관된 다음 레벨 타일들[도 6의 다음-레벨 타일들(52)]에 대해서 조합하고 따라서 후보 집합(102a, 102b) 내의 타일들의 수가 감소되는 것에 의해서, 후보 집합(102a, 102b)은 "컴팩트화"된다.(스텝 102는 또한 "컴팩트화"라고도 지칭된다). 여기에서 조합은 각각의 경우 4개 또는 그 이하의 타일들을 하나의 다음-레벨 타일에 조합하는 것에 의해서 달성된다. 만약 4개의 인접하는 타일들이 하나의 다음-레벨 타일로 조합된다면, 다음-레벨 타일이 커버하는 영역은 원래의 타일들과 동등하다. 만약 4개보다 작은 타일들이 조합된다면, 다음-레벨 타일이 커버하는 영역은 원래의 타일들의 영역을 전부 포함하는 상위 집합을 형성한다. 이러한 방식으로, 다음-레벨 타일들에 의해서 묘사되는 영역이 적어도 원래 타일들의 조합된 영역보다 크다는 점에서, 조합(컴팩트화)은 안전하고, 따라서 후보 집합(102a, 102b)에 연관된 타일들의 수는 적어도 커지게 되고, 즉 이후의 검색 스텝(스텝 105)에서, 더 많은 타일들이 그 내에서 포함된 객체들을 위해서 철저하게 검색되어야 한다. 그러나 컴팩트화에 의해서 임의의 정보도 손실되지 않으며, 즉 어떠한 진실적 후보도 배제되지 않는다.

[0106] 스텝 102에서의 컴팩트화는 후보 집합들(102a, 102b) 내의 타일들의 수를 감소시키는 것이 되고, 따라서 후보 집합들(102a, 102b)을 저장하기 위한 메모리 요구사항(requirements)을 감소시킨다. 후보 집합들(102a, 102b) 내의 타일들의 수는 이러한 방식으로 사전-정의된 임계값(예컨대 500)보다도 작은 수로 항상 제한된다.

[0107] 이렇게 획득된 후보 집합들(102a, 102b)로부터, 스텝 103에서는, 교차가 형성되어, 제1 후보 집합(102a) 및 제2 후보 집합(102b) 모두에 포함되는 것인 타일들만을 포함하는 조합 후보 집합(104a)을 산출하고, 그 결과로, 제1 검색 패턴 "KAP" 및 제2 검색 패턴 "WAR" 모두에 정합되는 속성을 가지는 객체들을 포함한다. 또한, 조합 후보 집합(104a) 내의 타일들의 수가 사전-정의된 임계값(스텝 102에서의 임계값과는 다를 수도 있음)을 초과하면, 조합 후보 집합(104a) 내의 타일들은 다시 컴팩트화될 수 있다(스텝 104).

[0108] 검색 쿼리에 정합하는 속성값들을 가지는 모든 객체들을 식별하기 위해서, 타일들의 조합 후보 집합(104a) 내에 포함된 타일들은 철저하게 검색된다(스텝 105). 즉 타일들은 하나씩 하나씩 조사(look through)되고, 모든 객체들은 검색 쿼리와 정합하는 속성값들이 추출된다. 그러한 결과적 객체들은 이후 출력 윈도우(2)를 통하여 출력된다. 예컨대, 출력 윈도우(2) 내에서 디스플레이되는 제1 객체는 거리 명칭, 즉 "WARstein"에서의 "KAPellenweg"이다.

[0109] 이러한 방식으로 발견될 수 있는 다른 객체들은 공항들, 관광지들 또는 이와 유사한 것과 같은 관심 지점이다.

이 문맥에서 언급되는 객체들은 내비게이션 장치 내에 포함되는 디지털 맵 데이터 내에서 식별 가능한 위치 배치를 가지는 것이고 또한 따라서 맵 상에서 위치 배치될 수 있다.

- [0110] 본 방법은 내비게이션 장치 내에서 처리되고 저장될 후보 집합들의 용이한 감소를 허용한다. 특히, 본 방법은 신속하고 효율적으로 다중-프레픽스 검색, 즉 특히 텍스트 스트링들과 같은 다중 검색 패턴들을 사용하고 검색 패턴들을 객체 속성들의 프레픽스들과 비교하는 것인 검색을 허용한다. 검색 패턴과 연관된 후보 집합을 발견하기 위해서, 검색 패턴의 모든 가능한 연속성들의 단위 집합이 형성되어야 한다. 트라이 구조 및 연관된 후보 집합들의 특정 지속성 묘사를 통하여, 실질적으로, 메인 메모리의 워크셋에 증분적으로 추가되는 원소들의 순차적 관독에 대한 프로세스를 감소시킨다.
- [0111] 이러한 것이 도 2에 도시된다. 도 2에 도시된 트라이(200)는 루트(201), 에지들(202) 및 노드들(nodes, 203)을 포함한다. 각각의 에지(202)는, 비공 스트링에 연관되고, 루트(201)로부터 노드까지의 길의 에지 라벨들의 연속에 의해서 라벨링된다. 예컨대 루트(201)에 대해서 왼쪽 아래에 있는 제1 노드는 "F"라고 라벨링되고, 그 노드 아래의 왼쪽에 있는 노드는 "FELDKIRCH"라고 라벨링되는 등이다.
- [0112] 도 2에 도시되듯이, 트라이(200) 내에서, 타일들은 파일(204) 내에서 그 타일 수들을 가지고 저장된다. 트라이(200)의 노드들(203)은 그러한 저장된 타일들의 컬렉션들(205, 206)에 연관된다. 예컨대, 컬렉션(205)은 "KAMEN", "KAMENER", "KAMENZ"로 라벨링된 노드들과 연관된다. 컬렉션(206)은 "KARLSRUHE"로 라벨링된 노드들과 연관된다.
- [0113] 트라이(200)의 형태로 인덱스 데이터를 구성하는 것은, 검색 패턴을 추가하는 것에 의해서 또는 기존 검색 패턴에 문자를 추가하는 것에 의해서 검색 패턴이 수정되는 것에 의해서 사용자가 검색 쿼리를 변화할 때, 한편으로는 신규 후보 집합(102a, 102b, 104a) 및 결과적 객체들의 대응하는 리스트를 효율적으로 연산하는 것을 허용한다. 이 경우에서 필요한 것은 단지, 이전에 획득된 조합 후보 집합(104a)과 신규 후보 집합의 교차를 형성하는 것인 단일 교집합 조작(도 1의 스텝 103)이다. 각 경우에서 획득된 조합 후보 집합(106)의 카디널리티는 감소되거나 적어도 증가되지는 않는다.
- [0114] 다른 한편으로는, 트라이(200) 구조는 신속하고 효율적인 방식으로 그 표현성에 관하여 입력된(또는 변화된) 검색 쿼리의 품질을 예측하는 것을 허용한다. 본 문맥에서, 표현성은 검색 영역을 국한하기 위한 검색 쿼리의 가능성을 결정한다. 검색 쿼리에 연관된 후보 집합이 작음(그러나 양인) 카디널리티를 가진다면, 검색 쿼리는 많은 표현성을 가진다. 여기에서는, 트라이(200)의 특정 지속성 구조는 집합의 합(sum)을 형성하지 않더라도 양호한 정확성으로 카디널리티를 예측하는 것을 가능하게 하고, 따라서 제한된 하드웨어 자원들을 가지는 모바일 장치들 상에서도 또한 해당 프로세스를 사용하는 것을 가능하게 한다. 이러한 방식으로, 검색 쿼리를 입력하는 동안에 검색 쿼리의 표현성에 대한 순간적(instant) 피드백이 사용자에게 주어질 수 있다.
- [0115] 도 3a 내지 도 3c는 내비게이션 장치의 입력 윈도우(1) 내로 검색 패턴의 입력 도중에서의 프로세스를 도시한다.
- [0116] 제1 스테이지에서, 입력 윈도우(1) 내로 "eif"를 입력한 후에는, 검색 쿼리의 표현성은 낮고, 이는 검색 쿼리에 연관된 가능한 결과적 객체들 및 타일들이 많으며 적절한(reasonable) 연산적 노력으로 철저한 검색이 수행되지 않는다는 것을 의미한다. 따라서, 출력 윈도우(2)는 비어 있고, 입력 윈도우의 배경 착색은 백색(또는 회색)이다.
- [0117] 사용자가 다른 문자를 입력하여 그의 검색 쿼리를 "eiff"로 변화하면, 검색 쿼리는 보다 표현성있게 되고, 즉 그에 연관된 타일들의 수가 감소한다. 적절한 연산적 노력으로 철저하게 검색될 수 있는 것인 후보 집합이 획득된다는 것을 지시하는 제1 사전-정의된 임계값보다, 만약 예측된 카디널리티가 작아진다면, 입력 윈도우의 착색은 노란색으로 변경된다. 신속하고 효율적으로 철저한 검색이 수행될 수 있다는 것을 지시하며 보다 더 작은 값인 제2 사전-정의된 임계값보다, 만약 예측된 카디널리티가 작아진다면, 입력 윈도우의 착색은 추가적으로 녹색으로 변경되고[포인트 셰이딩(pointed shading, 10)에 의해서 도 3b에서 지시됨], 획득된 후보 집합은 그 내에 포함된 모든 결과적인 객체들에 대해서 철저하게 검색된다. 발견된 결과적 객체들은 이후, 도 3b에 도시되듯이, 출력 윈도우(2) 내에서 출력된다. 만약 검색 쿼리의 입력이 검색 쿼리에 정합하는 속성들을 가지는 객체들을 포함하는 어떠한 타일들도 산출하지 않는다면, 후보 집합의 카디널리티는 0이 된다. 이것은 대부분의 경우에는 타이핑 오류에 기인하고, 시각적, 청각적 또는 감각적 경고(alarm)를 통하여 사용자에게 지시될 수 있다. 예컨대, 도 3c에서 도시되듯이 "eiff"를 입력할 때, 0의 카디널리티를 가지는 비어있는 후보 집합이 획득되고, [라인 셰이딩(11)에 의해서 도 3c 내에서 도시되는 바와 같이] 입력 윈도우의 적색 착색에 의해서 순간적으로 지시

되며, 가능하게는 장치의 진동들의 형태로 감각적 경고가 수반된다. 후보 집합이 비어 있으므로, 어떠한 결과적 객체들도 출력 윈도우(2) 내에서 디스플레이되지 않는다.

- [0118] 출력 윈도우(2) 내에서 결과적 객체들을 디스플레이할 때, 객체들의 순서는 사용자에게 대한 개연적 타당성에 따라서 선택된다. 이를 위해서, (예컨대 대도시들은 소도시들보다 보다 더 타당성있게 고려될 수 있다는 등의) 지리적 객체들에 연결되어 저장된 일반적 개연성 정보로부터 또는 사용자에게 의해서 입력된 검색 쿼리로부터 획득되는 정보를 사용하는 것인 휴리스틱 규칙들이 적용될 수 있다.
- [0119] (하나 또는 다중의 검색 패턴들로 구성된) 검색 쿼리의 입력은, 도 4a 내지 도 4c에 도시되듯이, 예컨대 내비게이션 애플리케이션이 구현되고 따라서 내비게이션 장치로서 기능하는 이동 전화의 숫자 키패드(3)를 통하여 수행될 수 있다. 키패드는, 공지되듯이, 12개의 키들, 즉 "0"에서 "9", "*", "#를 구비한다. 키 "*"는 입력 윈도우(1)에서 공백(space)을 삽입하기 위해서 사용되고, 키 "#"는 입력 윈도우(1)의 오른쪽 끝에서 숫자 또는 공백을(즉 가장 최근에 입력된 것을) 지우기 위해서 사용된다. 키들 중 일부에는 추가적으로 알파벳 문자들이 할당된다. 예컨대, 키 "2"(참조번호 31)에는, 또한 "ABC"라고 마킹되며, 이는 문자들 "A", "B", "C"가 추가적으로 해당 키에 대해서 할당된다는 것을 지시한다.
- [0120] 검색 쿼리를 입력할 때에, 키를 한번 이상 누르는 것이 필요하지는 않다. 예컨대, 문자 "C"에 대해서, 키 "2"는 단지 한번만 눌러질 수 있다. 따라서, 마치 전체 키보드를 사용하는 것처럼 키스트로크들의 복잡한 시퀀스들을 사용할 필요가 없이, 사용자는 검색 쿼리를 신속하고 용이한 방식으로 입력할 수 있다. 도 4a 내지 도 4c의 예시에서, 사용자는 숫자들 "5483776"를 하나씩 키패드(3)의 개별적인 키들을 누르는 것에 의해서 입력한다. 각각의 추가적인 숫자들을 입력하는 것으로, 검색 쿼리는 보다 표현성있게 되고, 출력 윈도우(2) 내에서 디스플레이되는 것인 결과적 객체들의 출력 리스트는 보다 짧아진다(도 4c에서 단지 3개의 객체들을 포함한다).
- [0121] 도 4a 내지 도 4c에 도시되듯이, (이동 전화의 일반적인 디스플레이일 수 있는) 디스플레이는, 예컨대 숫자로부터 가능한 알파벳 또는 알파벳 및 숫자[alphanumeric] 텍스트 스트링들로 변환하는(예컨대, 숫자들 "548377"을 대응하는 "LIVER7"로 변환하는, "7"은 연속성이 여전히 모호하다는 것을 지시함) 것인, 추가적인 윈도우(1')를 포함할 수 있다. 사용자가 더 많은 숫자들을 타이핑하여, 출력 윈도우(2) 내에 도시되는 정합하는 결과적 객체들의 수가 감소하면 할수록, 입력 윈도우(1) 내에 입력된 더 많은 숫자들이 모든 정합하는 검색 결과들 내의 하나 및 동일한 문자에 대응한다. 이러한 대응은 입력 윈도우(1) 하단의 윈도우(V)에 개별적으로 도시되지만, 또한, 가독성(readability)을 좋게 하기 위해서, 입력 윈도우(1) 내에 직접적으로 가시화될 수도 있다.
- [0122] 도 5a 내지 도 5g에 도시되듯이, 다른 실시예에서는, 입력 검색 쿼리에 대응하는 영역을 가시화하는 것인 지리적 영역이 도시될 수 있다. 도 5a 내지 도 5g의 예시에서, 두 개의 입력 윈도우들(1a 및 1b)이 존재하고 디스플레이되어, 하나의 입력 윈도우(1a)는 사용자가 구역에 대한 검색 패턴을 입력하도록 허용하고, 다른 하나의 입력 윈도우(1b)는 객체에 대한 검색 패턴을 입력하는 것을 허용한다. 영역 입력 윈도우 내에서 검색 패턴을 입력하는 동안(도 5b 내지 도 5d), 획득된 후보 집합은 점점 더 작은 카디널리티를 가지는 것으로 가정되고 또한 점점 더 적은 타일들을 포함한다. 따라서, 타일들을 철저하게 검색하는 것으로부터 획득되는 결과적 객체들의 수는 점점 적어지고, 도 5d에서는 구역 입력 검색 패턴에 대해서 단지 4개의 정합들만이 출력 윈도우(2) 내에서 디스플레이된다. 이제 객체 입력 윈도우(1b) 검색 패턴을 입력할 때(도 5e 내지 도 5g), 두 개의 입력 검색 패턴들 모두에 정합하는 것인 발견된 결과적 객체들이 출력 윈도우(2) 내에서 디스플레이되고, 검색 쿼리는, 각각의 입력 윈도우들(1a 및 1b) 내에서 더 많은 문자들을 입력하면 검색 쿼리는 보다 더 표현성을 가지게 된다.
- [0123] 입력 윈도우들(1a 및 1b) 내로 검색 패턴을 입력하는 동안에, 획득된 후보 집합의 타일들에 대응하는 지리적 영역을 도시하는 맵 세그먼트(4)가 입력 윈도우들(1a 및 1b) 다음에 디스플레이된다. 지리적 영역은 예컨대, 최북단, 최남단, 최동단 및 최서단 타일을 얻어서 그들 사이의 직사각형 지리적 영역에 걸쳐진다(spanning). 검색 쿼리가 보다 더 표현성있게 되면, 대응하는 지리적 영역도 점점 더 국한되어서, 디스플레이되는 맵 세그먼트(4)의 스케일도 점점 더 커지고, 따라서 관심 영역 내로의 클로징-인(closing-in)의 줌-유사(zoom-like) 인상을 가져온다.
- [0124] 도 5a 내지 도 5g의 시퀀스는 사용자가 검색 쿼리 "HAMBURG REEPERBAHN"의 문자들을 하나씩 입력하는 상황을 도시한다. 병행하여, 유럽 전체를 도시하는 맵 세그먼트(4)로부터 시작하여 "Reeperbahn"이라는 명칭의 거리가 위치 배치되는 함부르크 시의 일부의 클로즈-업에 도달할 때까지, 맵 세그먼트(4)의 스케일이 단계적으로 주밍된다. 맵 세그먼트(4)의 디스플레이 속도를 개선하기 위해서, 예컨대 숲 구역들과 같은 맵의 상세 사항들 중 덜 중요한 것들은 배제될 수 있다.

- [0125] 철저한 검색은 연산적으로 고가(expensive)이기 때문에, 타일들의 후보 집합 내에서 철저한 검색이 수행되기 전에, 각각의 경우, 바람직하게는, 맵 세그먼트(4)의 디스플레이는 갱신된다.
- [0126] 전술한 일반적인 방법의 개선을 포함하는 것인 데이터베이스 내에서 객체를 검색하는 방법의 다른 실시예의 흐름도가 도 7에 도시된다. 이 개선예에서는, 사용자 친화 및 연산적 속도 측면에서 본 방법의 효율성을 추가적으로 개선하기 위해서, 다중-페이지 휴리스틱 검색 접근법을 사용하며, 이후 상세하게 설명된다.
- [0127] 전술한 실시예들 뿐만 아니라, 이 개선예의 목적은, 큰 타일-구조 맵 데이터베이스로부터의 객체들의 작은 집합 또는 특정한 객체의 목표적 인출[targeted retrieval]을 지원하는 것이다. 여기에서는, 사용자 입력은 다중-프레픽스 검색 쿼리들의 구조화되지 않은(unstructured) 형태를 가지고, 따라서 교집합은 충분적이고 또한 응답적(responsive)이다. 이러한 기능성(functionality)은 따라서 네트워크 접속이 없고 제한된 자원들을 가지는 환경에서는 특별하게 공급되어야 한다.
- [0128] 여기에서의 "목표적 인출"이라는 용어는, 검색 엔진과의 상호 작용을 시작할 때, 사용자가 맵 데이터베이스로부터 특정한 객체(예컨대 도시, 거리 또는 관심 지점) 또는 객체들의 작은 집합을 의도하는 사실을 지칭한다. 이러한 객체는 또한 "사용자의 목표 객체"라고도 지칭될 수도 있다.
- [0129] 전술하듯이, 본 방법은, 예컨대 쿼트리의 타일들로 맵이 분할되는 것인(도 6 참조), 공간이 공간적 요소들로 분할된 데이터베이스를 사용한다. (맵 데이터베이스에서의 일반적인 경우와 같이) 데이터베이스가 크다면, 전체 데이터베이스를 철저하게 검색하는 것은 제한된-자원 환경에서는 비실용적일 수 있기 때문에, 본 방법에서는, 자원-제한된 휴리스틱 검색 접근법을 사용하며, 이의 검색 전략은 개략적으로 후술한다.
- [0130] 제1 페이지에서는(도 7에서 페이지 1로서 참조됨), 그 내에서 (철저한) 검색이 수행될 검색 공간은 특정한 휴리스틱 접근을 사용하여 감소된다. 이러한 휴리스틱 접근 내에서는, 어떤 휴리스틱 후보 집합들은, 사용자가 원하는 결과들을 포함하는 확률이 높은 검색 공간의 부분-집합들로서 선택된다. 이러한 (비교적 작은) 휴리스틱 후보 집합들은 크기가 증가하는 순으로(즉 가장 작은 후보 영역이 우선으로) 철저하게 검색된다.
- [0131] 최종 페이지(후에 상세히 설명되듯이, 제1 페이지와 최종 페이지 사이에는 더 많은 페이지들이 있을 수 있음)에서는, 곧, 어떠한 역할-특화 제약들도 사용하지 않고 확실하게 원하는 결과를 포함하여야 하는 방식으로, 전체 공간이 고려되고 일반적 후보 집합이 결정된다. 또한 캐치-올(catch-all) 후보 집합으로도 지칭되는 이 후보 집합은, 또한 철저하게 검색된다[역할-특화 휴리스틱 후보 집합들 중의 하나에서의 소속성(membership) 때문에 이미 철저하게 검색된 것은 이 검색에서 생략함].
- [0132] 자원 및 응답 시간의 제약에 부합(meet)하기 위해서, 만약 철저한 검색 페이지 도중에 자원 한계에 부딪히면 검색 프로세스는 중지되고, 애플리케이션은 이 상태(condition)를 통지받는다. 이후 예컨대 사용자가 보다 표현성이 있도록 쿼리를 확장하게 권장하며(예컨대 도 3을 참조하여 전술한 착색 스킴을 사용하여), 따라서 원하는 결과는, 만약 이번 라운드에서 이미 발견되지 않는다면, 다음 상호작용 사이클 내에서 원하는 발견될 수 있다. 전체 자원 한계(철저한 검색 내에서 고려되어야 할 맵 특징들의 최대 개수로 표현됨)는 검색 프로세스의 다양한 페이지들에 대해서 할당된 개별적인 쿼타들(quotas)로 분할되어서, 각각에게 공평한 기회를 주도록 한다(이러한 분할은 전용 휴리스틱들에 기반한다).
- [0133] 개선된 방법은 도 7을 참조하여 보다 상세하게 설명된다.
- [0134] 도 7에 도시되듯이, 사용자는 N개의 검색 단어들(검색 패턴들)로 구성된 검색 쿼리를, 본 방법이 실행되는 장치의 입력 윈도우(필드) 내에 입력한다(스텝 301). 스텝 302에서는, 곧, 입력 도중에 이미 또는 입력의 완성 직후에, 입력된 검색 쿼리의 서로 다른 해석 변형들이 생성되고 추가적인 검색 프로세스를 위해서 정리된다. 이러한 해석 변형들(또한 단순히 "해석들"로도 지칭됨)은, 감소된 검색 공간을 제공하기 위한 소위 역할-특화 휴리스틱 후보 집합들을 연산하기 위해서 사용된다.
- [0135] 역할-특화 휴리스틱 후보 집합들의 연산을 위한 시작점은 다음 관측점(observation)이다: 특정 목표 객체의 목표적 인출을 의도하는 특정한 검색 쿼리를 입력할 때, 이는 그의 의견으로는, 이 검색 쿼리에만 기반한 목표 객체들을 고유하게 식별하는 것이 원론적으로(in principle) 가능한 것이어야 하고, 또한 이는 (비록 항상은 아니지만) 종종 실제의 경우이다.
- [0136] 이제, 사용자가 상세하고 명쾌하게 생각하지는 않더라도, 사용자는 만약 질문을 받으면, 목표 객체를 명확하게 식별하기 위한 방식으로 어떻게 이러한 검색 쿼리가 이해될지를 보통 설명할 수 있다. 따라서 본 방법은 "단지" 이러한 해석을 "추측"하는 것만이 필요하다. 따라서, 맵 검색에 존재하는 다중 프레픽스 텍스트 쿼리들의 어떤

측면이, 매우 작은 쿼리들을 사용하더라도, 목표 객체들을 식별하는 것을 가능하게 하는 지에 대해서, 단지 정확한 해석이 사용된다면, 그리고 그러한 "정확한 해석들"이 어떻게 특성화될 수 있는지에 대해서, 조사된다. 이를 위해서, 검색 쿼리의 개별적인 입력 검색 패턴들은 특정한 역할을 가정한다고 간주될 수 있다. 이러한 역할들은 어떻게 검색 쿼리가 "의미되는지" 또는 "해석되어야 하는지"에 대한 기술과 관련된다. 검색 패턴을 위한 역할-특화 휴리스틱 후보 집합을 산출하는 것에 의해서, 이 역할들은 추가적으로 검색 공간의 현저한 감소를 허용할 수 있다. 역할-특화 휴리스틱 후보 집합은 역할 제약이 없는 동일한 검색 패턴의 후보 집합보다 더 작을 수 있다.

[0137] 만약 오퍼랜드들(operands)의 각각이(또는 적어도 대부분이) 역할 제약 때문에 더 작아진다면 감소 효과는 교집합 내에서 증폭되므로, 이러한 감소는 개별적인 검색 패턴의 역할-특화 휴리스틱 후보 집합들의 교집합으로서 연산한 조합 후보 집합들의 레벨에서 보다 효율적이다. 이러한 집합 크기 감소는, 역할 제약이 없는 후보 집합들을 사용하는 제약되지 않은 경우에 비교하여, 조합 휴리스틱 후보 집합이 증가된 공산(likelihood)을 가지면서 비어 있도록 할 수 있다.

[0138] "검색 패턴이 구역 명칭 내에서 발생한다." 또는 "검색 패턴이 객체 명칭의 첫 번째 단어이다."와 같이 검색 쿼리 내에서의 각각의 검색 패턴의 역할을 가정하는 것에 의해서, 다양한 검색 쿼리 해석 변형들이 획득된다. 명백하게, 검색 쿼리 내에서 더 많은 검색 패턴들이 구성된다면, 그러한 가정들의 더 많은 가능한 조합들이 존재한다.

[0139] 예컨대, 입력 검색 쿼리 "BER HUS"는, 첫 번째는 구역을 지칭하고 두 번째는 객체의 명칭을 지칭하는 2개의 입력 검색 패턴들 "BER" 및 "HUS"로 구성된다고 우선 해석될 수 있다. 다른 의미있는 해석은, 입력 검색 패턴이 객체의 명칭 내에서 그 순서대로 발생될 수 있다는 것이다. 다른 다중 해석들이 존재하고 고려될 수도 있다.

[0140] 주요 검색 프로세스가 시작하기 전에, 스텝 302에서는, 가장 특화된 해석들(역할-특화 휴리스틱 후보 집합들의 대부분이 매우 작은 것인)이 먼저 오는 방식으로, 조합에 관하여[combinatorially] 모든 의미있는, 비공 검색 쿼리 해석들이 열거되고 정리된다.

[0141] 전체 데이터베이스를 철저하게 검색하는 것이 보통 가능하지 않기 때문에, 이러한 해석들은 이후 휴리스틱 검색 프로세스가 어느 것을 채용할 지를 유도하는 데 사용된다. 각각의 해석은, 효율적으로 연산될 수 있는 것인 역할-특화 후보 집합들의 교집합으로서 정의되는 휴리스틱 후보 집합을 산출하고, 이러한 방식으로, 이러한 해석 변형들이 사용자의 의도를 정확하게 반영한다면 목표 객체에 포함되어야만 하는 것인 검색 공간의 작은 부분에 대해서 접근을 제공한다. 이러한 작은 부분은 이후 데이터베이스 객체들과의 정합을 실제로 식별하기 위해서 철저하게 검색된다.

[0142] 다음의 검색 패턴 역할들은 예컨대 스텝 302 내에서 검색 쿼리 해석들을 생성하기 위해서 사용될 수 있다:

[0143] A: 속성 카테고리들. 데이터베이스 속성들은 종종 특정한 카테고리에 속하는 것으로 특성화된다(characterized). 예컨대 맵 데이터베이스 내에서는, 객체는 하나의 명칭 속성과 여러 개의 구역-관련 속성들(객체가 위치하고 있는 국가 명칭, 도시 명칭, 구역 명칭들)을 가진다. 따라서 검색 공간 감소는, 각각의 검색 패턴이 특정한 특성 카테고리를 지칭한다고 간주하는 것에 의해서 달성될 수 있다. 예컨대 "첫 번째 단어는 명칭 속성을 지칭한다고 해석되고, 두 번째 단어는 구역 특성을 지칭한다."와 같이, 예컨대 각각의 사용자-제공 검색 패턴들에 대해서 카테고리를 할당하는 것에 의해서, 해석은 이후 (다른 것들 중에서) 특성화된다. 이러한 방식으로 조합에 관하여 가능한 카테고리 할당들의 조합들의 많은 수 중에서, 본 방법은 실용적(practical) 쿼리들에서 발생할 가능성이 높은 것인 추가적인 처리를 위한 조합들의 부분 집합을 선택할 수 있다. 최종적으로, 해석 변형은, 특정한 카테고리들을 지칭하도록 제약되지 않은 검색 패턴들에 대해서 고려된다.

[0144] B. 단어-순서(word-ordering) 제약을 가지는/가지지 않는 정합. 다음 두 개의 해석 변형들이 고려된다: 첫 번째로, 좀 더 특화된 해석으로서, 제1 검색 패턴은 정합되어야 할 객체의 명칭의 첫 번째 단어와 정합하여야 하고, 남아있는 검색 패턴들은 객체 명칭 내의 다른 단어들(첫 번째 위치에 있지 않는 것)과 정합하여야 한다. 다른 말로 하면, 검색 쿼리의 제1 검색 패턴은 "제1-단어 역할" 내에 있는 것으로 고려되고, 다른 검색 단어들은 "첫 번째가 아닌 단어 역할" 내에 있는 것으로 고려된다. 두 번째로, 보다 일반적인 해석으로서, 그들의 위치에 관계없이 각각의 검색 단어는 객체 명칭의 임의의 단어와 정합할 수 있다.

- [0145] C. 단어 수 제약을 가지는/가지지 않는 정합. 다음 두 개의 해석 변형들이 고려된다: 첫 번째로, 좀 더 특화된 해석으로서, 객체 명칭은 검색 쿼리로부터 도출될 수 있는 단어수를 가져야 한다. 다른 말로 하면, 각각의 검색 패턴들은 "k-단어 역할"로 할당되고, k는 검색 쿼리로부터 도출되는 수이다. 두 번째로, 보다 일반적인 해석으로서, 검색 패턴들의 수는 발견되어야 할 객체 명칭 내에서의 특정한 단어수를 암시한다고 이해되지는 않는다.
- [0146] D. 정확한/프레픽스 정합. 각 검색 패턴은 프레픽스로서(검색 패턴이 특정 검색 패턴으로 시작하는 객체 속성들과 정합함) 해석될 수 있거나 또는 정확한 템플릿(template)(객체 속성 내에서 정확하게 동일한 검색 패턴과만 정합함)로서 해석될 수 있다. 이 역할은 검색 패턴들 모두에 대해서 독립적으로 할당될 수 있기 때문에(그리고 다른 역할들에 독립적일 수 있기 때문에), 그리고 해석들의 조합에 관한 폭발적인 증가가 회피되어야 하기 때문에, 본 방법은 이러한 변형들의 선택된 부분 집합들에 대해서만 고려되도록 선택될 수 있다.
- [0147] 해석 변형들은 첫 번째로, 해당하는 역할-특화 휴리스틱 후보 집합들이 실제로 생성되기 전에, 추상적(abstract), 상징적 서술자들(descriptors)의 형태로 열거되고, 이후 철저히 스캐닝된다.
- [0148] 서로 다른 쿼리 해석들로부터 생성되는 휴리스틱 집합들의 크기는, 해석 내에서 사용되는 보다 일반적인 또는 보다 특화된 역할들 내의 정도(extent)에 대해서 강하게 관련(correlate)된다. 더 큰 것들 이전에 연산되고 스캐닝되는 더 작은 후보 집합들을 가지게 하기 위해서, 서술자들은 할당된 역할들을 고려하도록 평가(assess)된다. 이들은 이후 해당하는 휴리스틱 후보 집합들의 (예상되는) 크기가 증가하는 순(ascending order)으로 재-정리된다(re-ordered).
- [0149] 도 7에서 스텝 302를 완료한 후에, 입력된 검색 쿼리를 위한 서로 다른 해석 변형들은, 즉시 사용할 수 있고(at hand), 이후의 스텝들에서 특화 해석을 위한 휴리스틱 후보 집합을 연산하기 위해서 추가적으로 처리된다.
- [0150] 스텝 303에서는, 첫 번째로 전술한 정리에 따라서 가장 특화된 것으로 예상되는 해석이 고려된다. 스텝 304에서는, 이 해석을 위해서, (주요) 인덱스 데이터 구조 또는 보조적 인덱스 데이터 구조(객체의 구역 속성만을 예컨대 참조하는 것임) 내에서의 룩업을 통하고, 이후, 각각의 검색 패턴을 위해서(고려된 해석 내에서 할당된 특정한 역할에 대해서), 역할-특화 휴리스틱 후보 집합이 결정되고, 서로 다른 검색 패턴들의 역할-특화 후보 집합들을 교차하는 것에 의해서, 조합 휴리스틱 후보 집합이 획득된다(스텝 305). 가장 특화된 해석을 위해서, 이 휴리스틱 후보 집합은 (이 해석이 가장 특화된 것으로 예상되기 때문에 비록 가장 작은 것은 아니더라도) 개연적으로 작으며, 따라서 적어도 자원 한계에 부딪힐 때까지, 휴리스틱 후보 집합의 타일들은 철저히 검색될 수 있다(스텝 306). 결과들은 이후 서열화되어(ranked) 적절한 디스플레이 윈도우 내에 디스플레이된다(스텝들 307, 308).
- [0151] 이것은 고려되어야 할 모든 해석들에 대해서 반복되고, 최종 페이지(도 7의 오른쪽)에서는, 특정한 검색 패턴들이 특정한 역할들을 가지는 것으로 가정하는 것에 의해서 검색 쿼리가 제약되지 않는 것인 소위 캐치-올 교집합이 가정된다(스텝 309). 즉, (제약되지 않은) 후보 집합들이 전체 (주요) 인덱스 데이터 구조(트라이, 도 2 참조) 내에서의 룩업을 통하여 각각의 입력 검색 패턴을 위해서 획득되고 조합 후보 집합이 후보 집합들을 교차하는 것에 의해서 도출되는 점에서, 최종 페이지 내에서는, 전체 검색 공간이 고려된다(스텝들 310, 311). 이러한 조합 후보 집합은 또한, 자원 한계에 부딪힐 때까지 철저히 검색되고(스텝 312), 결과들은 서열화되고, 이전에 디스플레이되고(스텝 308) 획득된 결과들(스텝 307)과 합병된다.
- [0152] 그러한 단계들의 일부는 후속하여 보다 상세하게 설명된다.
- [0153] **휴리스틱 후보 집합 연산(스텝 304, 305)**
- [0154] 순간적인 방법에서는, 모든 후보 구역들이 타일 집합들(일부 경우는 컴팩트화된) 형태로 묘사된다. 데이터 구조의 설계에 의해서, 대단하지 않고[modest] 변함없는(constant) 양의 메모리를 취하면서도, 매우 효율적인 소속성 테스트, 포함(inclusion), 유니온 및 교차 조작들을 허용한다.
- [0155] 지속성 사전-연산된(pre-computed) 역할 특화 후보 집합들 내를 참조하는 지속성 인덱스들은, 검색 쿼리의 서로 다른 해석들로부터 후보 집합들의 전술한 시리즈들을 연산하는 것에 사용된다.
- [0156] 이 점에 대해서, 입력 검색 패턴의 서로 다른 "역할들"은 해당하는 휴리스틱 후보 집합을 결정하기 위해서 서로

다르게 취급될 수도 있다.

- [0157] 예컨대, "구역 카테고리 역할"로 검색 패턴이 할당될 때, 역할-특화 후보 집합을 연산하기 위해서, 예컨대 검색 단어들을, 그 내에서 객체들이 발생하며 검색 단어에 정합하는 구역 속성을 가지는 타일들의 집합들에 연관시키는 것에 보조적 인덱스 데이터 구조들은 사용될 수도 있다. 이러한 보조적 인덱스 데이터 구조는 또한 "구역 인덱스"라고도 지칭되고, 다른 인덱스 데이터 구조들과 마찬가지로 실제적인 검색 방법의 실행 이전에 사전-연산된다. 구역 인덱스 데이터는 도 2에 도시된 것과 구성 및 형태가 유사할 수 있지만, 타일들의 해당하는 컬렉션들에 객체의 구역 속성만을 연관한다(객체는 일반적으로 여러 가지 속성들로 기술되며, 예컨대 그 명칭과 그 구역이다).
- [0158] 구역 인덱스가 단지 구역 속성만을 타일들의 (집합들의) 컬렉션들에 연관하기 때문에, 결과적 후보 집합들 뿐만 아니라 구역 인덱스도, (주요) 인덱스 및 (주요) 인덱스 데이터 구조를 통한 스텝들로부터 도출되는 후보 집합들보다 크기 면에서 더 작다.
- [0159] 만약 입력 검색 패턴이 구역 명칭(구역은 객체의 속성임)을 지칭하는 것으로 해석된다면, 구역 인덱스는 해당되는 역할-특화 휴리스틱 후보 집합을 판독(read-out)하기 위해서 사용되고 스텝핑된다(steped). 예컨대, 만약 입력 검색 패턴 "BER"이 구역 속성을 지칭하는 것으로 해석된다면, 구역 인덱스에 의해서, 이 특정한 검색 패턴에 해당하는 타일들의 컬렉션이 어떤 것인지 결정되고, 따라서 원하는 역할-특화 휴리스틱 집합을 결정한다. 구역 속성은 단지 하나의 객체 속성이기 때문에, (주요) 인덱스 데이터 구조를 사용하여 임의의 객체 속성에 대해서 입력 검색 패턴을 정합하는 것에 의해서 획득될 수 있는 후보 집합에 비해서, 획득된 역할-특화 후보 집합은 더 작아진다.
- [0160] 보조적 인덱스 데이터 구조들을 채용하는 것 대신에, 또는 추가적으로 그러한 보조적 인덱스 데이터 구조들을 사용하는 것 대신에, (주요) 인덱스 데이터 구조를 확장하여 역할-특화 휴리스틱 후보 집합들을 결정하는 것도 가능하다. 이러한 확장을 위해서, 전용 추가적 브랜치들(branches)이 트리-형태 인덱스 데이터 구조에 부가될 수도 있다. 이러한 방식으로, 예컨대, 입력 검색 패턴을 첫 번째 단어로써 고려한 객체 명칭들에 해당하는 타일들의 컬렉션인 (주요) 인덱스 데이터 구조 내에서 추가적으로 사전-저장될 수 있다. 예컨대 입력 검색 패턴 "BER"은 객체의 명칭 내에서의 첫 번째 단어로써 해석될 수 있고("첫 번째 단어 역할"), 이 입력 검색 패턴에 그들의 명칭의 첫 번째 단어로써 "BER"을 가지는 객체들로 구성된 것인 타일들만이 연관된다.
- [0161] 저-표현성(low-expressiveness) 검색 패턴들을 위해서, 후보 집합들(특히 역할-제약되지 않은 경우)은 다음과 같은 두 가지 이유들 때문에 크고 연산에 많은 시간이 걸린다: 타일 수들의 많은 수가 지속성 타일 집합들로부터 판독되어야 하고, 후속하는 근사적 타일 집합 구조로의 메모리-내 응집(in-memory aggregation) 동안에, 컴팩트화가 빈번하게 수행되어야 하기 때문이다.
- [0162] 그러나, 이러한 문제가 발생하는 비표현성인 인덱스 내에서는 단지 비교적 작은 수의 프레픽스들(서유럽 맵 데이터베이스를 예로 하는 경우 수천 개)만이 존재한다.
- [0163] 따라서, 이러한 문제들의 경우에 대해서, 사전-컴팩트화되고, 효율적으로 로딩 가능한(loadable) 집합들을 제공하는 것인, 분리된 전용 지속성 "벌크"(bulk) 인덱스 및 집합 조합이 채용될 수 있다. 집합 특업은 우선 벌크 인덱스 내에서 엔트리를 발견하도록 시도하며, 만약 이것이 실패하면, 일반적 인덱스 및 연관된 지속성 타일 집합들이 사용된다.
- [0164] 각각의 역할-특화 휴리스틱 후보 집합의 실제적인 연산은 해당하는 해석 변형의 상징적 묘사에 기반한다. 변형에 따라서, 서로 다른 (주요 및 보조적) 인덱스들이 지속성 집합 묘사들로부터 특정한 성분(component) 집합들을 인출하기 위해서 접속된다. 지속성 집합 묘사들은, 실제적인 휴리스틱 후보 집합들을 획득하기 위해서, 이후 메모리 내로 로딩되고 조합된다(교차된다).
- [0165] 실용적인 검색 세션들 내에서는, 조합 후보 집합들의 연산을 위해서 요구되는 후보 집합들의 많은 웨어(share)가 한 번 이상 필요하다. 연산된 후보 집합들을 위한 캐싱(caching) 메커니즘이 따라서 이러한 집합들을 재-연산하기 위한 노력들을 대부분의 시간에서 절약하는 것을 보증한다.
- [0166] **철저한 검색 및 스트링 정합(스텝 306)**
- [0167] 후보 집합이 연산되면(그리고 비공이면) 바로, 철저하게 검색된다(즉 특정한 해석을 위한 스텝 306 내에서 그리고 제약되지 않은 경우를 위한 스텝 312에서). 스트링 정합은, 각각의 검색 패턴이 문자-대-문자(character-by

character)로 일부의 데이터베이스 객체 속성과 정합하여야 하는 것인 정규의 "정확한 모드" 내에서 또는 검색 패턴들과 객체 속성 사이에서 특정한 "허용되는 편차들(deviations)"이 존재하더라도 정합이 등록되는 것인 "근사적 모드" 내에서 발생할 수 있다. 허용되는 편차들의 일부 예시(일반성을 제한하지 않음)는:

- [0168] - 비록 객체 속성에서는 원래의 구별적 문자(예컨대, "Ö" 가 사용되더라도, 검색 패턴 내에서의 검색 패턴 내에서 구별적 문자의 "기본 문자"(예컨대 "O")를 사용하는 것,
- [0169] - 비록 데이터베이스 속성은 원래의 특정한 문자(예컨대 "ß")를 포함하더라도, 검색 패턴 내에서의 특정한 문자의 문자 바꿈(transliteration)(예컨대 "ss")을 사용하는 것
- [0170] - 실제적인 데이터베이스 속성("EIFFEL TOWER") 대신에, 원래 검색 패턴이 잘못 타이핑되었으나 인식 가능한 변형(예컨대 "EIFEL TOWER")을 사용하는 것)
- [0171] 전체에 대한 철저한 검색 프로세스는, 예컨대 정합들의 수에 대한 애플리케이션-특화 한계가 도달하는 것과 같은 자원-기인된(resource-oriented) 한계들이 도달되거나 또는 충분한 정합이 발견되는 즉시 중단된다.
- [0172] 그러한 후보 집합들(또는 도한 역할-특화 휴리스틱 집합들의)의 연산 도중에 부정확한(inexact) 정합을 지원하기 위해서, 본 방법은 데이터베이스 객체들의 속성 내에서 발생하는 원래 단어들을 포함하는 것 뿐만 아니라 또한 전술한 바와 같이 이러한 단어들의 변형들을 포함하는 것인 "강화된"(enriched) 사전-연산된 인덱스 데이터 구조들을 사용할 수도 있다. 본 방법은 또한, 철자가 잘못 되었으나 인식 가능한 검색 패턴들이 제시된 때, 인덱스 데이터 구조로부터 데이터베이스 객체 속성 내에 발생하는 원래 단어를 인출하기 위해서, 첫 번째로 인덱스 내에 존재하지 않는 하나 또는 그 이상의 유사한 단어들을 식별하고, 이후 타일들의 정확한 연관된 컬렉션들을 인출한다.
- [0173] 본 방법은 또한 이러한 두 가지 접근들의 조합을 사용할 수도 있다.

[0174] 결과 서열화(스텝 307)

- [0175] 사용자는 결과적 리스트에서 단지 목표 객체만을 보는 것을 원하는 것이 아니라 또한 결과적 리스트의 최상부 근처의 것도 보는 것을 기대한다. 이를 달성하기 위해서, 다양한 지시자들(indicators)(일부 경우는 본문 내에서 "신호들"이라고도 지칭됨)을 사용하여 각각의 결과의 타당성이 평가된다. 각 지시자는, 평가된 결과 데이터베이스 객체의 일부 특성(property)이 이러한 객체가 사용자에게 타당한 지 또는 어느 정도로 타당한 지를 나타내는 것인 숫자값이다. 이러한 특성들 및 이에 연관된 지시자들은 다음과 같이 카테고리들로 분류될 수 있다:
- [0176] 의미적 데이터베이스-내재 특성들[Semantic database-inherent properties]: 특정한 데이터베이스 객체들은 다른 것들보다도 평균적인 사용자에게 타당한 것인 더 높은 개연성을 가진다. 이러한 클래스의 지시자들은, 데이터베이스 내용(content)에 관한 의미적 지식을 사용하여, 단지 데이터베이스 객체 자체만을 평가한다. 예컨대, 대도시들 또는 거리들은 작은 것들에 비해서 더 중요하게 서열화된다.
- [0177] 쿼리-정합 특성들: 이 카테고리의 지시자들은, 의미적 지식을 사용하지 않고 어휘적(lexical) 및 구문적(syntactical) 레벨 상에서 조작된다. 대신에, 이들은 쿼리가 어떻게 데이터베이스 객체들을 정합하는 지 상세하게 평가한다. 예컨대, 정확한 정합은 부정확한 정합보다도 더 중요하게 서열화될 수 있다.
- [0178] 최종 서열화는 모든 지시자들의 가중 조합[weighted combination]이다. 본 방법은, 결과적 리스트를 사용자에게 제시하기 전에, 이러한 서열화에 의해서 암시되는 정리를 사용하거나 또는 추가적인 재-정리 기술들[예컨대 특정한 공통 특질들(traits)에 따라서 결과들을 클러스터링(clustering)하는 것]을 적용할 수 있다.
- [0179] 후속하여, 본 방법의 추가적인 개선예에서 사용될 수 있는, 일부 추가적인 개념들(concepts)이 설명된다.

[0180] 추가적인 개념들

- [0181] 다수의 데이터베이스에 대한 검색: 종종 사용자가 검색을 모든 데이터가 하나의 동일한 데이터베이스 내에서 유

열화 스킴을 사용하여 단일 결과로 저장된 결과들을 통합하는 것에 의해서, 확장된다. 개별적인 검색 프로세스들은 전술한 형태이거나 또는 공통된 서열화 페이지에서 처리될 수 있도록 결과가 적응되는(adapted) 완전히 다른 외부 검색 프로세스들일수도 있다.

- [0182] 이러한 것이 도 8에 가시화된다. 여기에서는, 그래픽적 서술[graphical depiction]은 수직 박스들에 의해서 지시되는 개별적인 검색 프로세스들의 특화된 리를 암시하는 것을 의미하지는 않는다.
- [0183] 도 8의 예시에서는, N개의 입력 검색 패턴들로 구성된 검색 쿼리가 입력된다(스텝 401). 스텝들 402 내지 405에서는, 입력된 검색 쿼리에 기반하여, 검색들이 서로 다른 데이터베이스 내에서 수행된다. 스텝 402는 예컨대 전술한 유형(type)의 검색 방법을 사용한 데이터베이스 내에서의 검색에 해당할 수 있다. 스텝 403은 유사한 검색 방법을 사용한 다른 데이터베이스 내에서의 검색에 해당할 수 있다. 스텝들 404 내지 405는 완전히 다른 검색 방법들을 사용한 또 다른 데이터베이스 내에서의 검색에 해당할 수 있다.
- [0184] 스텝들 406 및 407에서는, 단계들 402 및 403의 결과들에 호환되도록(compatible)(예컨대 그 포맷에서), 외부 검색 프로세스들의 결과들이 적응되고, 따라서 결과들은 스텝 408 내에서 서열화되고, 등록되고, 합병되며, 스텝 409 내에서 함께 디스플레이된다.
- [0185] 검색 프로세스들(스텝들 402 내지 405)은 병렬적으로 또는 그 순서가 애플리케이션에 적응될 수 있는 것으로 순차적으로 실행될 수 있다.
- [0186] 소스 데이터베이스들의 쿼리-표현성 유도된 선택[Query-expressiveness guided selection of source databases]:
- [0187] 사용자 입력 도중의 초기에, (부분적, 불완전적) 검색 쿼리는, 그 낮은 표현성 때문에, 큰 데이터베이스 내에서 어떠한 의미있는 검색 공간 감소를 위해서는 충분하지 않다. 이 상황 내에서 실제적 정합들을 열거하는 것은 가능하나, 이는 사용자에게는 쓸모없고 혼란스럽게 비춰질 것이다. 쿼리의 표현성이 그에 적용될 데이터베이스의 크기에 관한 것이므로, 단지 데이터베이스의 보다 더 작은, 선택된 부분(데이터베이스의 세그먼트)만을 이 상황에서 고려하는 것에 의해서 또는 다중-데이터베이스 검색에서 사용가능한 소스 데이터베이스들의 부분집합만을 고려하는 것에 의해서, 이는 완화(alleviate)될 수 있다.
- [0188] 일부 디폴트(default) 데이터베이스 소스들(또는 세그먼트들)을 제외하면, 만약 쿼리의 표현성(이 데이터베이스에 관하여 고려되거나 또는 근사적으로 묘사되는 것으로 선택되는 일부 소스 데이터베이스에 관하여 고려됨)이 특정한 임계값을 넘어선다면, 각각의 사용가능한 소스 데이터베이스(또는 데이터베이스 세그먼트)는 단지 검색 프로세스 내에서 포함된다. 내비게이션 애플리케이션 내에서, 만약 쿼리의 표현성이 특정한 임계값보다 작다면, 검색 프로세스는 예컨대 다음 소스들 상에서 조작될 수 있다.
- [0189] - 단지 중요한 도시들 및 공항들이 고려되는 것인 맵 데이터베이스 세그먼트
- [0190] - 사용자의 선호 목적지 리스트
- [0191] - 특정한 맵 객체들에 대한 약어들(abbreviations)의 보충적(supplementary) 데이터베이스(예컨대, 당국에서 차량 번호 등록판 상에 사용하는 코드들로 축약된 도시들)
- [0192] 따라서, 검색 쿼리 스트링의 길이 그 자체(per se)는 검색 프로세서 내에서, 검색 스트링의 표현성이 아니라, 데이터베이스 소스들 또는 세그먼트들의 포함에 관하여 결정하는 데 사용된다. 표현성은 쿼리 길이에 엄격하게(strictly) 관련되지는 않는다: 데이터베이스 내에서 거의 발생하지 않는 "XYZ"와 같은 짧은 쿼리는 그럼에도 불구하고 표현성이 높으며, "BAHNHOFSTRASSE"와 같이 상당히 긴 쿼리는, 이러한 명칭의 거리가 수천 개의 도시들에서 존재하기 때문에, 상대적으로 낮은 표현성을 가진다. 여기에서 설명하는 본 방법은 주어진 검색 단어의 표현성의 예측을 효율적으로 획득하는 것을 가능하게 한다 - 이는 그의 (일반적인)후보 집합의 크기에 역비례하여 관련된다.
- [0193] 자주-선택된 선호 결과들의 자체-학습 신속 검색(Self-learning fast search of frequently-selected favorite results): 예컨대 내비게이션 애플리케이션 내에서 사용자가 자주 방문하는 목적지들에 해당하는 것인, 사용자가 상대적으로 자주 검색하는 데이터베이스 객체들이 있을 수 있다. 그러한 객체들의 인출을 위한 상호 작용을 축소하기 위해서, 다음 방법이 사용된다:

- [0194] 사용자가 결과적 리스트로부터 객체를 선택하고 애플리케이션 내에서의 추가적인 처리를 위해서 넘겨줄(hand over) 때(예컨대 내비게이션 애플리케이션 내에서 목적지를 선택하고 경로 계산을 시작할 때)마다, 선택된 객체는 선호 결과들의 지속성으로 저장된 리스트 내에 등록된다. 이 리스트는 사용자-당(per-user) 기반으로 저장되고, 만약 주변 시스템 또는 애플리케이션이 다중 사용자들을 지원한다면, 다중의 지속성 리스트들이 존재할 것이다. 이 리스트는 데이터베이스 객체들에 대한 참조, 또는 목표 객체를 고유하게 특성화하는 결과적 속성들(검색 결과 리스트 내에서 일반적으로 사용됨)을 저장할 것이다.
- [0195] 선호 리스트는 이후 일반화된 검색 프로세스에 외부 데이터베이스로서 통합되고, 전술한 바와 같이 표현성-유도된 데이터베이스 소스 선택 프로세스에 의해서 검색 입력의 초기 스테이지들부터 포함된다.
- [0196] 단어-레벨 퍼지 정합 및 연관된 검색: 전술한 검색 공간 축소 기술들은 타일 세트들의 레벨 상에서 조작(operate)된다. 이것은, 쿼리 내의 각각의 검색 패턴에 대해서, 이 패턴에 정합하는 일부의 객체를 포함하는(그리고 이러한 후보 집합을 산출하는 해석을 위해서 역할 특화되어 수행됨) 후보 집합이 보증된다는 것을 의미한다. 그러나 동시에 모든 입력 검색 패턴들에 정합하는 하나의 객체가 존재한다는 것을 보증하지는 않는다. 실제적인 정합들은 스트링 정합 페이즈 및 철저한 검색 내에서 식별되며, 매우 자주, 이러한 페이즈 동안에 객체들은 단지 입력 검색 패턴들의 일부에만 정합하게 된다.
- [0197] 이러한 불완전한 정합들을 무시하는 대신에, 이들은 다음과 같이 취급될 수 있다:
- [0198] - 만약 일부 추가적인 요구들을 만족한다면(예컨대 객체의 명칭 속성 상에서 적어도 하나의 정합이 존재하여야 한다는 것), 비-정합(non-matching) 단어들의 특정한 수는 수용 가능한(acceptable) 것으로 고려된다.
- [0199] - 불완전한 정합들은 결과적 리스트 내에서 불완전성의 정도(즉, 정합을 산출하지 못한 검색 단어들의 수)를 가지고 마킹(mark)된다.
- [0200] - 불완전한 정합들이 임의의 발견된 완전한 정합들도 "제치 두지"(push aside) 않도록 하는 방식으로, 이러한 정도는 결과 서열화 내에서 고려된다.
- [0201] - 결과적 리스트들을 제시할 때, 불완전한 정합들은 특별하게 마킹되어, 사용자가 즉시 이들을 인식하고 이들이 완전하지 않다는 것을 알 수 있다.
- [0202] 이러한 기술은 다음의 기본적인 연관 검색 기능을 제공한다: 만약 사용자가, 그 조합이 예컨대 임의의 개별적인 데이터베이스 객체와도 완전하게는 정합하지 않지만 동일한 맵 타일 내에서 위치하는 두 개의 객체들을 참조하고 이러한 조합이 단지 맵 타일들의 작은 집합 또는 하나에서만 발생하는 것인, 두 개의 검색 패턴들을 입력하면, 검색 공간 축소 페이즈 동안에 연산되는 후보 집합들 중 하나는 이러한 타일을 포함하고, 단어-레벨 퍼지 정합은 참조된 객체들 모두를 식별하고 또한 이들을 단어-레벨 퍼지 결과들로서 등록한다.
- [0203] 이에 의해서, 예컨대, 쿼리 "NEUE OSTKR"에서 "NEUE BAHNHOFSTRASSE"라는 거리를 인출하는 것이 가능하다. 이 객체는 입력 검색 패턴 "OSTKR"에 정합하지는 않지만, 독일 베를린의 기차역 "OSTKREUZ"과 동일한 타일 내에서(따라서 근처인) 위치 배치된다.
- [0204] 추가적인 개선예에서, 본 방법은, 조합 후보 집합을 연산하기 전에 인근의 타일들을 포함하는 것에 의해서, 개별적인 후보 집합들을 수정(amend)한다. 이것은 연관된 검색에서, 하나의 동일한 타일 내에서 위치 배치되지는 않지만 두 개의 인근의 타일들에서 위치 배치되는 객체들의 조합을 발견하는 것을, 가능하게 한다.
- [0205] 대안적인 입력 방법들:
- [0206] 검색 특징들을 가지는 음성 인식 또는 이와 함께 필기 인식과 같은 입력 방법을 사용하는 것은 서로 다른 애플리케이션 문맥 내에서 매력적일 수 있다. 그러한 입력 방법들 및 연관된 소프트웨어는 일반적으로 사용자의 발언(utterance)의 텍스트 묘사를 생성하고, 이는 전술한 유형의 텍스트 검색 방법들에서 사용될 수 있다.
- [0207] 여기에서 설명된 방법들을 사용하여, 집합 연산들의 사용은 그러한 입력 방법들의 통합을 위한 추가적인 방안들(avenues)을 가능하게 한다: 이러한 방법들이 개연적 방법들에 일반적으로 기반하고 있으므로, 이 방법들은 단지 하나의 출력 텍스트들만이 아니라 서열화된 변형 리스트들을 도출(deliver)할 수 있다.

- [0208] 도 9에 예시되듯이, 사용자는 음성 입력을 통하여, 두 개의 단어들을 입력하고, 시스템은 가장 서열이 높은(highest-ranked) 결과로서, "Köln"(슬롯 1) 및 "Bremerhaven"(슬롯 2)으로 해석한다. 여기에서 각각의 단어는 소위 슬롯에 할당된다. 이 경우에는, 모호하지 않게 사용자의 음성 입력을 식별하는 것이 가능하지 않기 때문에, 2번째 서열이 높은(second-ranked) 결과는 "Möln"(슬롯 1) 및 "Römerhaven"(슬롯 2)이다. 음성 입력 시스템이 결정할 개연성은, 슬롯 1의 결과들에 대해서는 0.7 대 0.3이고, 슬롯 2의 결과들에 대해서는 0.6 대 0.4이다.
- [0209] 각각의 슬롯에서 최선으로 서열화된 결과는 실제적인 사용자의 발언을 묘사한다고 보증할 수는 없고, 서로 다른 슬롯으로부터의 임의의 조합은 실제 발언된 검색 쿼리를 묘사한다고 할 수 있다.
- [0210] 본원에서 전술한 방법들을 채용한 후보 집합 연산들은 도 9에 도시되는 바와 같이 가능한 조합들의 다수를 용이하게 배제할 수 있다: 슬롯들의 쌍에 대해서 (유한한) 후보 집합들을 교차하는 것은 비어 있는 집합(점선 화살표로 도시됨)을 산출할 수도 있고, 이는 데이터베이스 내에서 이러한 검색 패턴들의 조합과 어떠한 정합도 존재하지 않는다는 것이다. 이것은, 단지 후보 집합들을 연산하고 교차하는 것에 의해서, 가능한 슬롯 조합들의 절반이 제거된다는 것을 의미한다. 두개의 비공 조합후보 집합은 서로 다른 퀄리티의 두 개의 정합을 포함한다: 도 9에서 "KÖLN RÖMERHAFEN"이 정확한 정합이다(도 9에서 실선 화살표로 도시됨). "KÖLN BREMERHAVEN"은 도 9에 대시-화살표로 도시되듯이 단지 프레픽스 정합(BREMERHAVENER STRASSE)을 산출한다. 만약 사용자가 항상 프레픽스들보다는 완전한 단어들로 발언한다고 기대되면, 집합 연산은 "정확한" 역할을 사용하는 역할-특화 후보 집합들을 사용하여 수행될 수 있다. 여기에서, 조합 "'KÖLN BREMERHAVEN"은 비어 있는 결합 후보 집합을 산출한다. 즉 모든 부정확한 조합들도 집합 연산 레벨 상에서 제거된다.
- [0211] 일반적으로, 이러한 방식으로 추가적인 고려로부터 슬롯 엔트리들의 조합의 상당한 수가 용이하게 제거될 수 있어서 매우 낮은 연산적 비용을 가진다.
- [0212] 수퍼-인덱스의 사용: 여기에서 설명한 검색 방법은, 미디어 데이터의 매우 큰 컬렉션들에 대한 검색-기반 접근을 용이하기 위하여, 메타데이터가 본 검색 방법을 사용하여 인덱스되는 것인, 미디어 재생 장치(media player) 또는 셋탑 박스 애플리케이션들에 대해서 사용될 수도 있다. 미디어 데이터가 자연적인 지리적 내장(embedding)을 가지지 않기 때문에, 이들은 인위적(artificial) 타일 수로 할당될 수 있다. 미디어 재생 장치들 및 셋탑 박스들의 원격 제어기는 종종 매우 제한된 키보드들을 가지며, 따라서 간략화 키보드들을 사용하여 전술한 섹션들 내에서 설명된 기술들이 장점을 가질 수 있다.
- [0213] 가능하게는, 이 문맥 내에서의 개선예에서는, 미디어 재생 장치로 이동되어야 할 필요가 있는 일부 또는 모두의 메타데이터 정보는 중앙 데이터베이스로부터 도출될 수 있다. 이것은, 출시되어서 (대부분의 경우) 글로벌 데이터베이스들 내에 등록되는 것인 콤팩트 디스크들로부터의 일부 방식으로 대부분의 경우 기인하는 것인 예컨대 뮤직 타일들에 대해서 수행될 수도 있다. 따라서 글로벌 데이터베이스로부터 수퍼-인덱스를 구축하고 이 수퍼-인덱스를 장치 상에 실제로 인스톨된 미디어 파일들의 작은 부분 집합을 검색하기 위해서 사용하는 것이 가능하게 된다. 전제 조건(prerequisite)으로서, 국부적으로 인스톨된 파일들은 반드시 등록되어야 하고, 따라서 주어진 타일 수에 연관된 파일들이 용이하게 접속될 수 있어야 한다. 이러한 것은 이들을 단순한 트라이-형태 인덱스 구조 내로 정렬하는 것에 의해서 수행될 수 있다. 인위적 타일 수를 연산한 이후에, 미디어 파일의 발견가능성(findability)은 수퍼-인덱스를 이용하여 검사된다: 미디어 파일의 메타데이터로부터 구축된 검색 쿼리는 인위적 타일 수를 포함하는 후보 집합을 산출하여야만 한다. 만약 이러한 경우가 아니라면, 미디어 파일은 글로벌 데이터베이스에서도 알지 못하는 것이 된다. 이와 유사한 경우들에서는, 작은 보조적 데이터베이스가 장치 상에서 유지되고 검색 프로세스 내로 통합된다.
- [0214] 본 발명은 전술한 실시예들에 한정되지 않는다. 예컨대, 본 발명은 2차원 평면 내에서의 내비게이션을 구현하는 것들에 한정되지 않으며, 예컨대 우주항공 또는 항공 애플리케이션들과 같이 쿼드트리나 아인 옥트트리에 의해서 분할된 3차원 공간에도 적용될 수 있으며, 또는 1차원 검색 공간(예컨대 전자-책 내의 페이지들)에도 적용될 수 있다.
- [0215] 위에서 미디어 재생 장치 시나리오에 대해서 이미 설명한 바와 같이 본 방법은 또한 원래 데이터베이스 객체들이 공간적 참조들을 전혀 가지지 않는 애플리케이션 내에서도, 원래 데이터들의 인위적 공간적 내장을 생성하는 것에 의해서, 즉 각각의 데이터 객체에 k-차원 공간 내에서의 합성(synthetic) 위치를 할당하는 것에 의해서,

사용될 수 있다.

[0216] 본 방법은 또한, 쿼리 입력, 결과 제시 및 검색 프로세스의 실행이 하나의 동일한 컴퓨팅 장치에서만 수행될 필요가 없이 다른 양방향 또는 단방향 통신 링크들 및/또는 네트워크를 통하여 연결된 다수의 컴퓨팅 장치들 사이에서 분산될(distributed) 수도 있는 것인 네트워크형 또는 커넥티드 시나리오 내에서도 적용될 수 있다. 하나의 예시는, 웹브라우저 내에서 디스플레이되는 웹 페이지 상에 표시되는 입력 필드 내에서 검색 쿼리가 사용자에게 의해서 입력되고 이후 검색 쿼리가 본 방법이 실행되는 서버와 통신하고 결과들이 네트워크를 통하여 통신되고 상기 웹 페이지의 동적(dynamic)인 부분 내에서 표시되는 것인 인터넷-기반 애플리케이션이다. 다른 예시는 적외선 통신 링크를 사용하는 원격 제어를 구비하는 미디어 센터 장치이며, 채널들 또는 미디어 파일들에 부착된 텍스트 서술들 및 명칭들을 검색하는 것을 가능하게 하기 위하여, 본 방법이 미디어 센터 장치 상에서 실행되며, 원격 제어 상의 키패드가 사용자에게 의해서 입력 쿼리들을 입력하기 위해서 사용되며, 미디어 센터 장치에 부착된 TV 세트가 결과들을 사용자에게 디스플레이하기 위해서 사용되는 것이다.

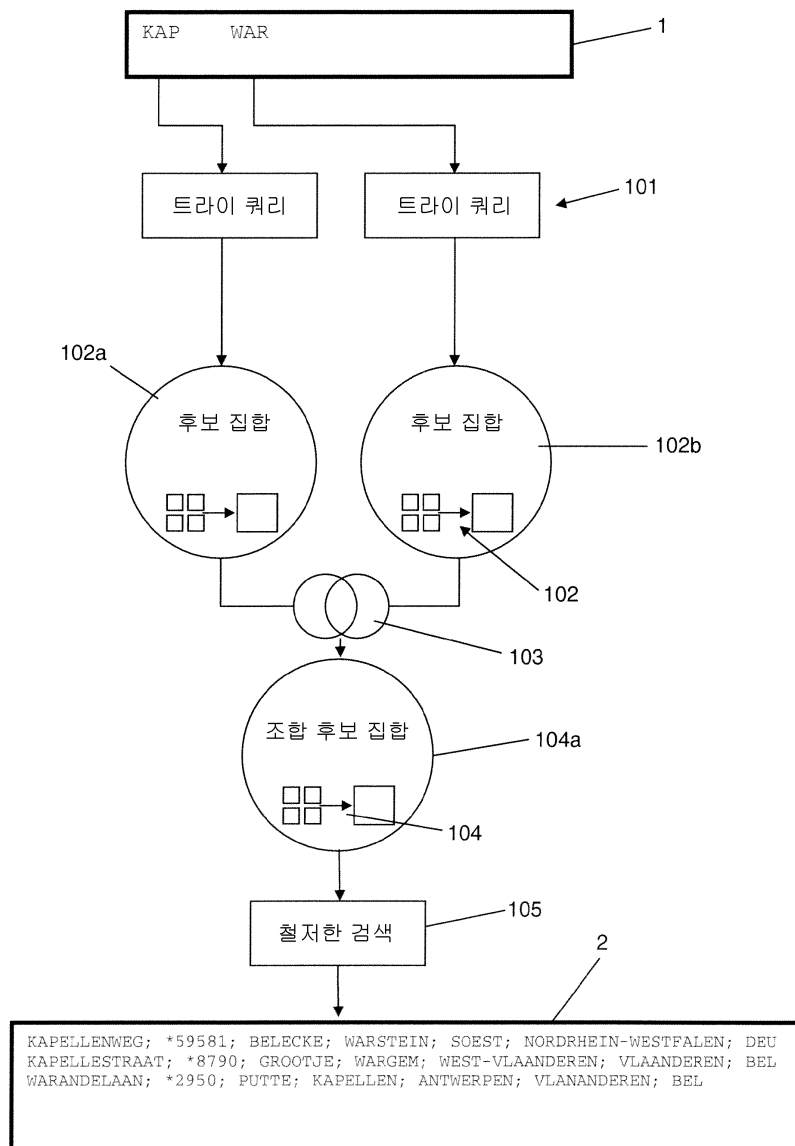
부호의 설명

[0217]

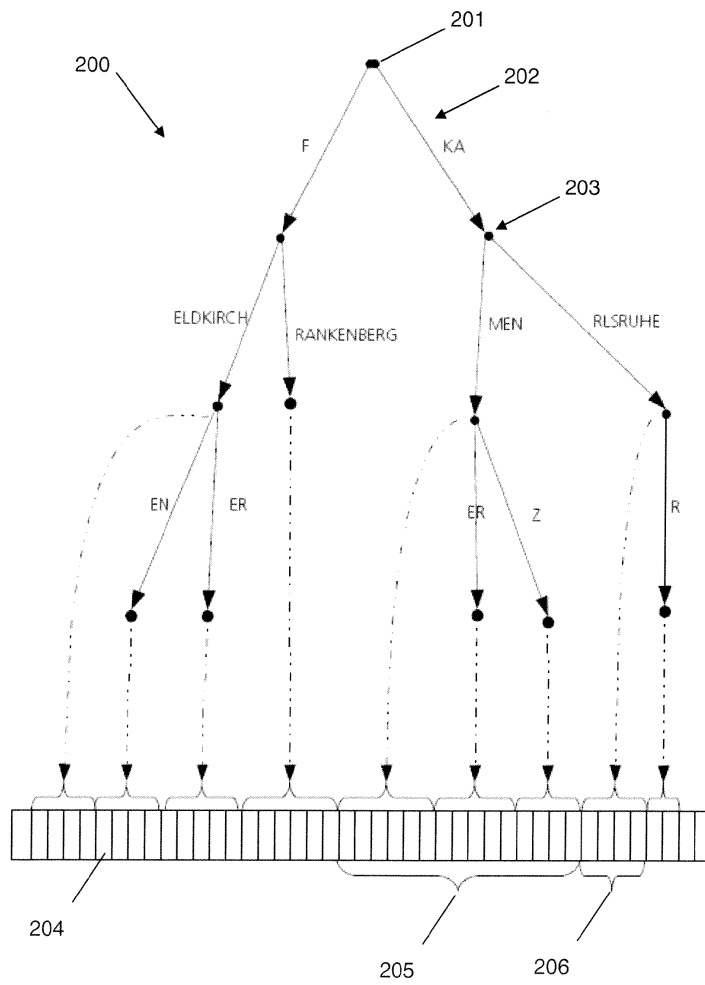
1 , 1a, 1b: 입력 윈도우	1': 윈도우
10, 11: 셰이딩	2: 출력 윈도우
3: 키패드	31: 키
4 맵 세그먼트	5: 맵
51: 맵 타일	52: 다음-레벨 타일
101-105: 스텝들	102a, 102b: 후보 집합
104a: 조합 후보 집합	200: 트라이
201: 루트	202: 예지들
203: 노드들	204: 타일 수들의 파일
205: 공간적 원소들의 컬렉션	
206: 공간적 원소들의 컬렉션	
301-312: 스텝들	401-409: 스텝들

도면

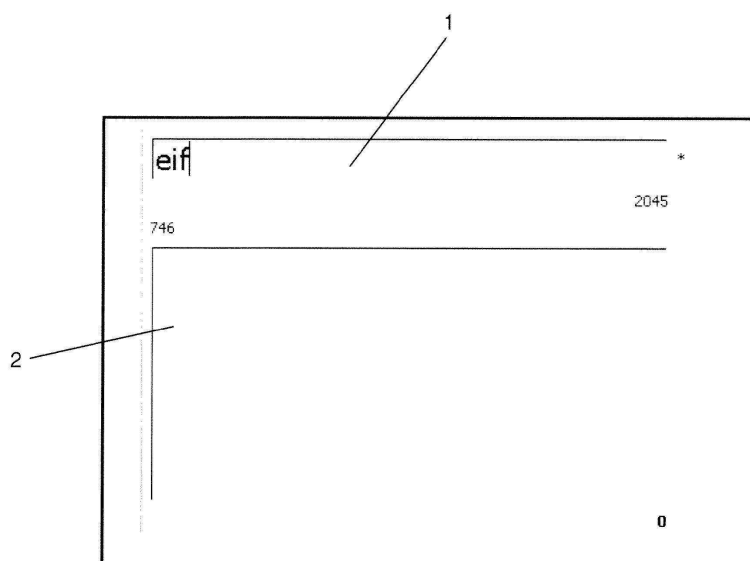
도면1



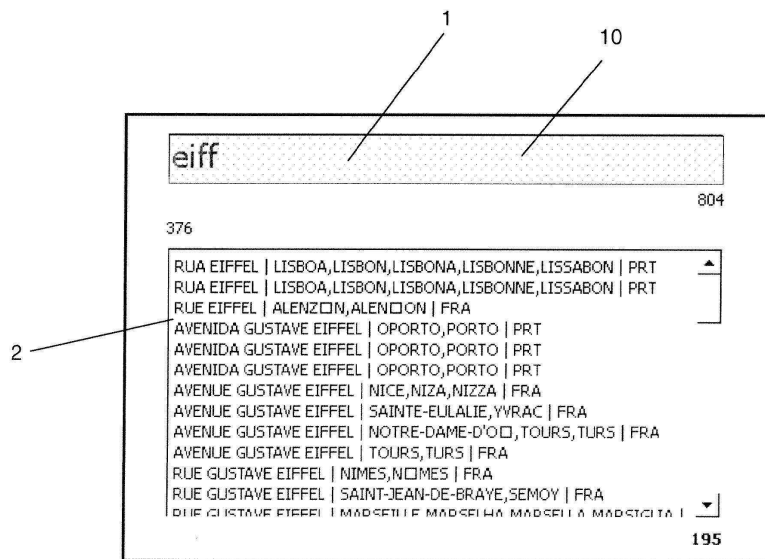
도면2



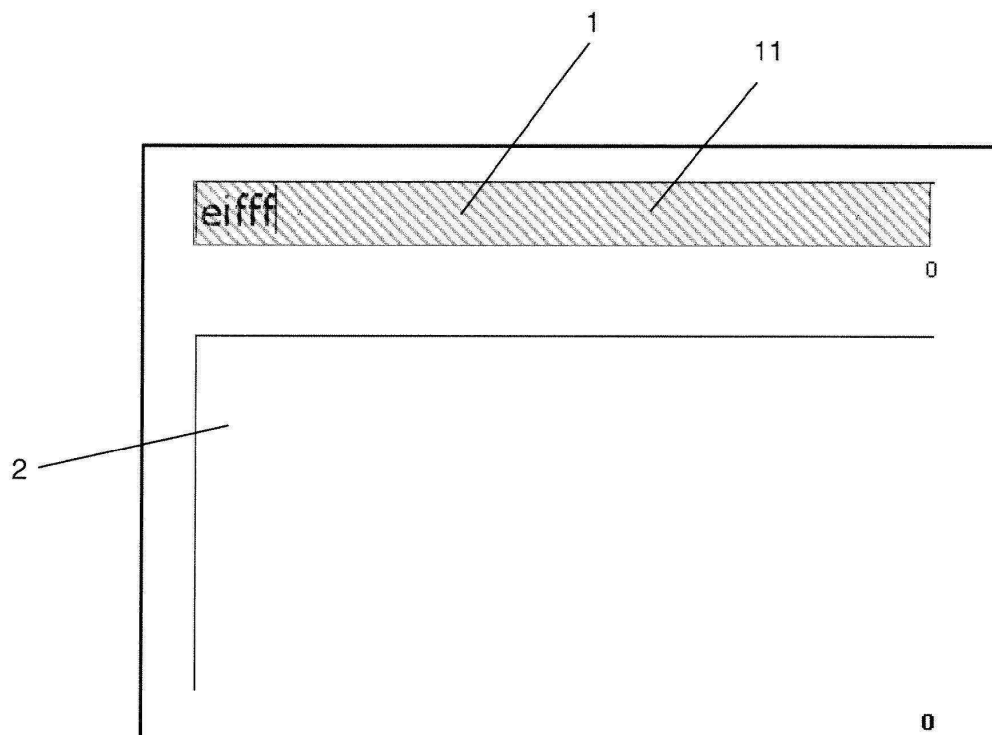
도면3a



도면3b



도면3c



도면4a

Figure 1 is a schematic diagram of a data entry interface. The interface consists of a main window with a title bar (1) and a menu bar (1'). The window contains a text area (2) with a list of airport names, a search field (3) with the text '54837', and a results table (31) with columns for airport name, IATA code, and ICAO code. The results table lists airports like Tamarit de Litera, Tamarit de Litera, Liverpool, Lennon Airport, etc.

도면4b

The diagram shows a data entry interface with the following elements and annotations:

- Annotation 1:** Points to the top horizontal line of the first input field.
- Annotation 1':** Points to the top horizontal line of the second input field.
- Annotation 2:** Points to the left vertical line of the third input field.
- Annotation 3:** Points to the top horizontal line of the fourth input field.
- Annotation 31:** Points to the top horizontal line of the fifth input field.

The interface consists of several input fields and labels:

- Field 1: Contains the text "548377".
- Field 2: Contains the text "1682".
- Field 3: Contains the text "LIVER7".
- Field 4: Contains the text "LIVERPOOL AIRPORT#ENG LIVERPOOL JOHN LENNON AIRPORT | LIVERPOOL AIRPORT,LIVERPOOL JOH | GBR | WERS-CAZELLES | LA BONNETTID | FRA".
- Field 5: Contains the text "LIVERPOOL | LIVERPOOL | GBR".
- Field 6: Contains the text "LIVERSEDGE | LIVERSEDGE | GBR".
- Field 7: Contains the text "SAINT-JEAN-DE-LIVERSAY | SAINT-JEAN-DE-LIVERSAY | FRA".
- Field 8: Contains the text "1".
- Field 9: Contains the text "2 ABC".
- Field 10: Contains the text "3 DEF".
- Field 11: Contains the text "4 GHI".
- Field 12: Contains the text "5 JKL".
- Field 13: Contains the text "6 MNO".
- Field 14: Contains the text "7 PQRS".
- Field 15: Contains the text "8 TUV".
- Field 16: Contains the text "9 WXYZ".
- Field 17: Contains the text "*<spc>".
- Field 18: Contains the text "0".
- Field 19: Contains the text "#<del>".

There are also some additional labels and symbols:

- A "*" symbol is located between the first and second input fields.
- A "*" symbol is located between the third and fourth input fields.
- A "*" symbol is located between the fourth and fifth input fields.
- A "*" symbol is located between the fifth and sixth input fields.
- A "*" symbol is located between the sixth and seventh input fields.
- A "*" symbol is located between the seventh and eighth input fields.
- A "*" symbol is located between the eighth and ninth input fields.
- A "*" symbol is located between the ninth and tenth input fields.
- A "*" symbol is located between the tenth and eleventh input fields.
- A "*" symbol is located between the eleventh and twelfth input fields.
- A "*" symbol is located between the twelfth and thirteenth input fields.
- A "*" symbol is located between the thirteenth and fourteenth input fields.
- A "*" symbol is located between the fourteenth and fifteenth input fields.
- A "*" symbol is located between the fifteenth and sixteenth input fields.
- A "*" symbol is located between the sixteenth and seventeenth input fields.
- A "*" symbol is located between the seventeenth and eighteenth input fields.
- A "*" symbol is located between the eighteenth and nineteenth input fields.

도면4c

Diagram illustrating a terminal window layout for a flight reservation system. The window is divided into several sections:

- Header Section:** Contains the flight number **5483776** and the status **158**.
- Flight Details Section:** Labeled **LIVERPO**, it displays the following text:


```
LIVERPOOL AIRPORT:#ENG LIVERPOOL JOHN LENNON
AIRPORT | LIVERPOOL AIRPORT,LIVERPOOL JOH | GBR
LIVERPOOL | LIVERPOOL | GBR
```
- Control Grid:** A grid of buttons for navigation and editing, labeled 1 through 9, and special function buttons:

1	2 ABC	3 DEF
4 GHI	5 JKL	6 MNO
7 PQRS	8 TUV	9 WXYZ
* <spc>	0	# <del>

Annotations in the diagram:

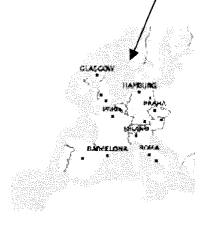
- 1** points to the flight number **5483776**.
- 1'** points to the status **158**.
- 2** points to the flight details section.
- 3** points to the control grid.
- 31** points to the button labeled **3 DEF**.

도면5a

목적지 선택

구역: 1a 1b

객체:



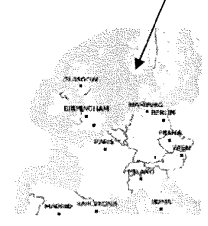
2

도면5b

목적지 선택

구역: 1a 1b

객체:



2

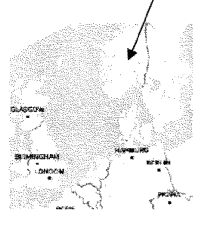
HAMBLEDEN, RG9 6 HENLEY ON THAMES, GBR
 HAMBLETON, FY6 9 POULTON LE FYLDE, GBR
 ALT-HAMBORN, 47166 DUISBOURG, DEU
 HAMBURG-INSEL NEUWERK, 27499 AMBURGO, DEU
 HAMBÜHREN, 29313 HAMBÜHREN, DEU
 HAMBURG, 27729 HAMBURG, DEU
 HAMBÜCKEN, 76707 HAMBÜCKEN, DEU
 HAMBACH, 57910 HAMBACH, FRA

도면5c

목적지 선택

구역: 1a 1b

객체:



2

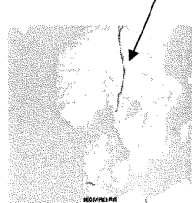
HAMBURG, 20354 AMBURGO, DEU
 HAMBURG-INSEL NEUWERK, 27499 AMBURGO, DEU
 HAMBÜHREN, 29313 HAMBÜHREN, DEU
 HAMBÜCKEN, 53809 RUPPCHTEROTH, DEU
 HAMBÜCKEN, 53809 RUPPCHTEROTH, DEU
 HAMBÜCKEN, 53809 RUPPCHTEROTH, DEU
 HAMBÜCKEN, 53809 RUPPCHTEROTH, DEU
 HAMBÜCKEN, 53809 RUPPCHTEROTH, DEU
 HAMBÜCKEN, 53809 RUPPCHTEROTH, DEU
 HAMBÜCKEN, 53809 RUPPCHTEROTH, DEU

도면5d

목적지 선택

구역: hamburg

객체:



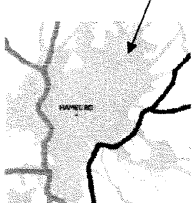
HAMBURG, 20354 AMBURGO, DEU
HAMBURG-INSEL NEUWERK, 27499 AMBURGO, DEU
KLEIN HAMBURG, 25557 GOKELS, DEU
HAMBURGLANDET, 7268 FRØYA, NOR

도면5e

목적지 선택

구역: hamburg

객체: ree



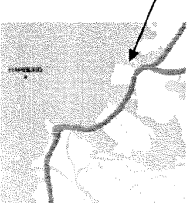
AM REE, 22459 AMBURGO, DEU
ANTON REE SCHULE, 21035 AMBURGO, DEU
ANITA-REE-STRASSE, 21035 AMBURGO, DEU
ANTON-REE-WEG, 20537 AMBURGO, DEU
REEPERBAHN, 20359 AMBURGO, DEU
REEBORN, 22415 AMBURGO, DEU
REEKAMP, 22415 AMBURGO, DEU
REEMBRODEN, 22339 AMBURGO, DEU
REEMBUSCH, 22179 AMBURGO, DEU

도면5f

목적지 선택

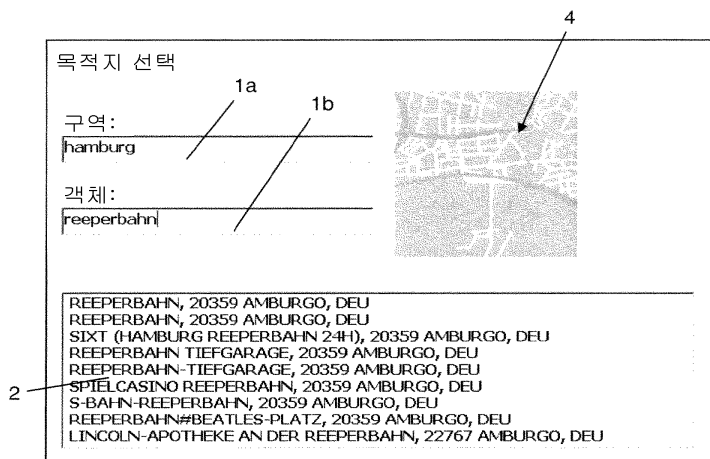
구역: hamburg

객체: reeper

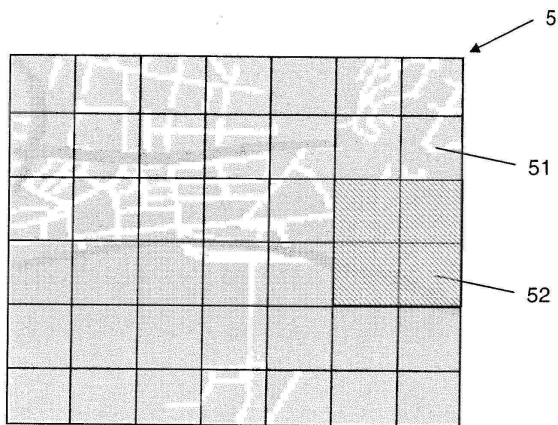


REEPERBAHN, 20359 AMBURGO, DEU
REEPERBAHN, 20359 AMBURGO, DEU
REEPERSTIEG, 21029 AMBURGO, DEU
SIXT (HAMBURG REEPERBAHN 24H), 20359 AMBURGO, DEU
REEPERBAHN-TIEFGARAGE, 20359 AMBURGO, DEU
REEPERBAHN-TIEFGARAGE, 20359 AMBURGO, DEU
SPIELCASINO REEPERBAHN, 20359 AMBURGO, DEU
S-BAHN-REEPERBAHN, 20359 AMBURGO, DEU
REEPERBAHN#BEATLES-PLATZ, 20359 AMBURGO, DEU

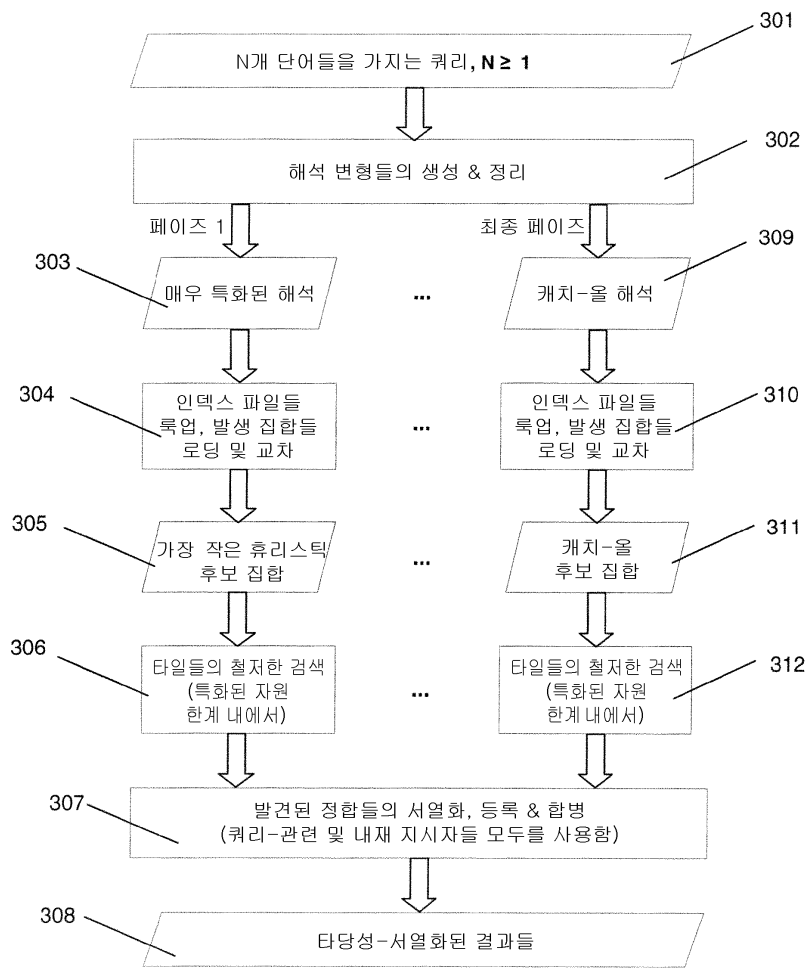
도면5g



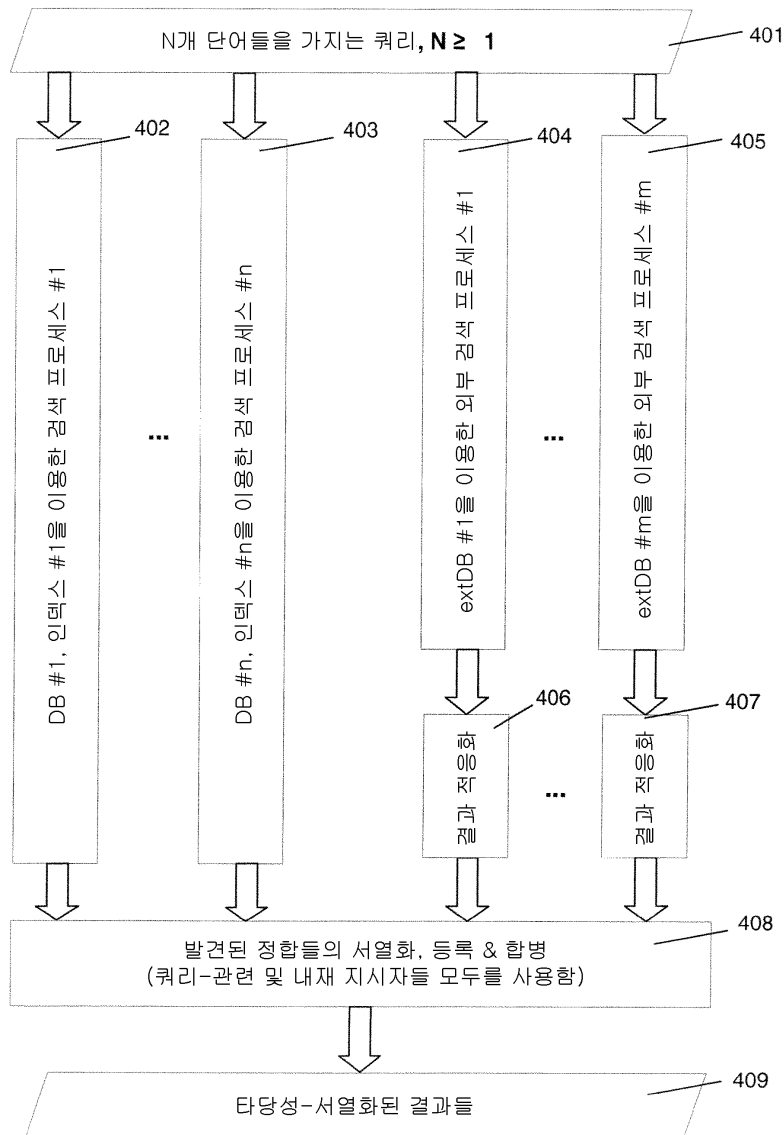
도면6



도면7



도면8



도면9

슬롯 1		슬롯 2
0.7 KÖLN		0.6 BREMERHAVEN
0.3 MÖLLN		0.4 RÖMERHAFEN