

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5237837号
(P5237837)

(45) 発行日 平成25年7月17日(2013.7.17)

(24) 登録日 平成25年4月5日(2013.4.5)

(51) Int.Cl.

F I

G 0 6 F 17/30 (2006.01)

G 0 6 F 17/30 4 1 4 A

G 0 6 F 17/30 1 7 0 C

請求項の数 10 (全 27 頁)

(21) 出願番号 特願2009-1234 (P2009-1234)
(22) 出願日 平成21年1月7日(2009.1.7)
(65) 公開番号 特開2010-160591 (P2010-160591A)
(43) 公開日 平成22年7月22日(2010.7.22)
審査請求日 平成23年11月1日(2011.11.1)

(73) 特許権者 000005108
株式会社日立製作所
東京都千代田区丸の内一丁目6番6号
(73) 特許権者 000233055
株式会社日立ソリューションズ
東京都品川区東品川四丁目12番7号
(74) 代理人 100064414
弁理士 磯野 道造
(74) 代理人 100111545
弁理士 多田 悦夫
(72) 発明者 林 秀樹
東京都国分寺市東恋ヶ窪一丁目280番地
株式会社日立製作所 中央研究所内

最終頁に続く

(54) 【発明の名称】 空間データ管理装置、空間データ管理方法、および、空間データ管理プログラム

(57) 【特許請求の範囲】

【請求項1】

多次元空間上にオブジェクトが配置される空間データを管理する空間データ管理装置であって、

前記空間データ管理装置は、空間データを格納する記憶部と、指定された検索範囲内に存在するオブジェクトを前記空間データから検索するオブジェクト検索部と、を有し、

前記記憶部には、各ノードからの子ノードの数を制限するバランス木として前記空間データを格納する領域索引ツリーが記憶され、

前記記憶された領域索引ツリーの各ノードには、前記空間データを1回以上等分割した空間領域の情報が所定数以下だけ格納され、その空間領域の情報は、空間領域の多次元空間上の位置を一意に示すスカラ値に写像した結果であるキー値が含まれ、

前記領域索引ツリーの子ノードを持つ非リーフノードには、その各子ノードのとりうるキー値の範囲に応じて、各子ノードへのリンクが対応づけられており、前記領域索引ツリーの子ノードを持たないリーフノードには、空間領域の情報から、その空間領域に外接矩形が交わるオブジェクトへのリンクが対応づけられており、

前記オブジェクト検索部は、

入力装置を介して前記検索範囲が指定されると、

前記領域索引ツリーのルートノードから順に、前記検索範囲の代表点をスカラ値に写像した検索符号値と、前記領域索引ツリーの各ノードの空間領域のキー値とを照合することにより、非リーフノード、リーフノード、オブジェクトへのリンクと辿ることで、前記検

10

20

索範囲の代表点が含まれる空間領域と、その空間領域からリンクするオブジェクトとを抽出する手順について、一巡ごとに前記検索範囲から辿った空間領域を除外した領域の代表点を新たな検索符号値として更新することで繰り返し、

前記検索範囲から辿った空間領域を除外した領域が無くなると、前記領域索引ツリーの各ノードを辿る処理により抽出される各オブジェクトについて、オブジェクトの外接矩形と前記検索範囲とを照合し、少なくとも一部が交わるオブジェクトを、前記検索範囲に交わるオブジェクトとして出力することを特徴とする

空間データ管理装置。

【請求項 2】

前記空間データ管理装置は、前記領域索引ツリーのキー値および前記検索範囲の代表点を示す検索符号値をそれぞれ示す空間データ上のスカラ値として、モートン符号値を計算することを特徴とする

請求項 1 に記載の空間データ管理装置。

【請求項 3】

前記空間データ管理装置は、前記領域索引ツリーのキー値および前記検索範囲の代表点を示す検索符号値をそれぞれ示す空間データ上のスカラ値を格納するためのメモリ容量として、空間領域の多次元空間上の位置を一意に示すスカラ値のとりうる値を表現するための最小限の容量を決定することを特徴とする

請求項 1 または請求項 2 に記載の空間データ管理装置。

【請求項 4】

前記オブジェクト検索部は、さらに、オブジェクトの外接矩形と前記検索範囲とで少なくとも一部が交わるオブジェクトについて、前記記憶部に格納されているオブジェクトの実形状と、入力されている前記検索範囲の実形状とを照合し、少なくとも一部が交わるオブジェクトを、前記検索範囲に交わるオブジェクトとして出力することを特徴とする

請求項 1 ないし請求項 3 のいずれか 1 項に記載の空間データ管理装置。

【請求項 5】

空間データを格納する前記記憶部には、前記領域索引ツリーからリンクされる各オブジェクトについて、そのオブジェクトの種別を示す種別 ID が対応づけられており、

前記オブジェクト検索部は、検索条件として前記検索範囲および種別 ID が指定されると、前記領域索引ツリーから辿ることで抽出されるオブジェクトの内、検索条件の種別 ID と一致するオブジェクトを選択して、出力することを特徴とする

請求項 1 ないし請求項 4 のいずれか 1 項に記載の空間データ管理装置。

【請求項 6】

空間データを格納する前記記憶部には、複数の空間領域に対応する複数の前記領域索引ツリーが記憶されており、

前記オブジェクト検索部は、検索条件として前記検索範囲が指定されると、その前記検索範囲が示す空間データ上の位置情報をもとに、複数の前記領域索引ツリーから 1 つの前記領域索引ツリーを選択することを特徴とする

請求項 1 ないし請求項 5 のいずれか 1 項に記載の空間データ管理装置。

【請求項 7】

前記空間データ管理装置は、さらに、前記領域索引ツリーにオブジェクトを登録するオブジェクト登録部を有し、

前記オブジェクト登録部は、前記入力装置を介して登録対象のオブジェクトが指定されると、

登録対象のオブジェクトの位置情報を基に、登録対象のオブジェクトが交わる前記領域索引ツリーの空間領域を 1 つ以上特定し、その特定した空間領域の情報を格納する各ノードから、登録対象のオブジェクトへのリンクを対応づけ、

1 つの空間領域からリンクされるオブジェクトが所定数を越えたときには、その空間領域を等分割することで、複数の空間領域へと領域分割して前記領域索引ツリーを更新することを特徴とする

10

20

30

40

50

請求項 1 ないし請求項 5 のいずれか 1 項に記載の空間データ管理装置。

【請求項 8】

前記空間データ管理装置は、さらに、前記領域索引ツリーに登録されたオブジェクトを削除するオブジェクト削除部を有し、

前記オブジェクト削除部は、

前記入力装置を介して削除対象のオブジェクトが指定されると、前記領域索引ツリーの各ノードからリンクされているオブジェクトのうち、削除対象のオブジェクトを削除し、

削除したオブジェクトのリンク元である前記領域索引ツリーの各ノードについて、隣接する等分割された複数の領域からそれぞれリンクされるオブジェクトが所定数を下回ったときには、それらの空間領域を 1 つの空間領域へと領域統合して前記領域索引ツリーを更新することを特徴とする

10

請求項 1 ないし請求項 6 のいずれか 1 項に記載の空間データ管理装置。

【請求項 9】

多次元空間上にオブジェクトが配置される空間データを管理する空間データ管理装置による空間データ管理方法であって、

前記空間データ管理装置は、空間データを格納する記憶部と、指定された検索範囲内に存在するオブジェクトを前記空間データから検索するオブジェクト検索部と、を有し、

前記記憶部には、各ノードからの子ノードの数を制限するバランス木として前記空間データを格納する領域索引ツリーが記憶され、

前記記憶された領域索引ツリーの各ノードには、前記空間データを 1 回以上等分割した空間領域の情報が所定数以下だけ格納され、その空間領域の情報は、空間領域の多次元空間上の位置を一意に示すスカラ値に写像した結果であるキー値が含まれ、

20

前記領域索引ツリーの子ノードを持つ非リーフノードには、その各子ノードのとりうるキー値の範囲に応じて、各子ノードへのリンクが対応づけられており、前記領域索引ツリーの子ノードを持たないリーフノードには、空間領域の情報から、その空間領域に外接矩形が交わるオブジェクトへのリンクが対応づけられており、

前記オブジェクト検索部は、

入力装置を介して前記検索範囲が指定されると、

前記領域索引ツリーのルートノードから順に、前記検索範囲の代表点をスカラ値に写像した検索符号値と、前記領域索引ツリーの各ノードの空間領域のキー値とを照合することにより、非リーフノード、リーフノード、オブジェクトへのリンクと辿ることで、前記検索範囲の代表点が含まれる空間領域と、その空間領域からリンクするオブジェクトとを抽出する手順について、一巡ごとに前記検索範囲から辿った空間領域を除外した領域の代表点を新たな検索符号値として更新することで繰り返し、

30

前記検索範囲から辿った空間領域を除外した領域が無くなると、前記領域索引ツリーの各ノードを辿る処理により抽出される各オブジェクトについて、オブジェクトの外接矩形と前記検索範囲とを照合し、少なくとも一部が交わるオブジェクトを、前記検索範囲に交わるオブジェクトとして出力することを特徴とする

空間データ管理方法。

【請求項 10】

40

多次元空間上にオブジェクトが配置される空間データを管理する空間データ管理装置による空間データ管理方法を、コンピュータである前記空間データ管理装置に実行させるための空間データ管理プログラムであって、

前記空間データ管理装置は、空間データを格納する記憶部と、指定された検索範囲内に存在するオブジェクトを前記空間データから検索するオブジェクト検索部と、を有してあり、

前記記憶部には、各ノードからの子ノードの数を制限するバランス木として前記空間データを格納する領域索引ツリーが記憶され、

前記記憶された領域索引ツリーの各ノードには、前記空間データを 1 回以上等分割した空間領域の情報が所定数以下だけ格納され、その空間領域の情報は、空間領域の多次元空

50

間上の位置を一意に示すスカラ値に写像した結果であるキー値が含まれ、

前記領域索引ツリーの子ノードを持つ非リーフノードには、その各子ノードのとりうるキー値の範囲に応じて、各子ノードへのリンクが対応づけられており、前記領域索引ツリーの子ノードを持たないリーフノードには、空間領域の情報から、その空間領域に外接矩形が交わるオブジェクトへのリンクが対応づけられており、

前記オブジェクト検索部に、

入力装置を介して前記検索範囲が指定されると、

前記領域索引ツリーのルートノードから順に、前記検索範囲の代表点をスカラ値に写像した検索符号値と、前記領域索引ツリーの各ノードの空間領域のキー値とを照合することにより、非リーフノード、リーフノード、オブジェクトへのリンクと辿ることで、前記検索範囲の代表点が含まれる空間領域と、その空間領域からリンクするオブジェクトとを抽出する手順について、一巡ごとに前記検索範囲から辿った空間領域を除外した領域の代表点を新たな検索符号値として更新することで繰り返し、

前記検索範囲から辿った空間領域を除外した領域が無くなると、前記領域索引ツリーの各ノードを辿る処理により抽出される各オブジェクトについて、オブジェクトの外接矩形と前記検索範囲とを照合し、少なくとも一部が交わるオブジェクトを、前記検索範囲に交わるオブジェクトとして出力する処理を実行させるための

空間データ管理プログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、空間データ管理装置、空間データ管理方法、および、空間データ管理プログラムに関する。

【背景技術】

【0002】

近年、二次元、または三次元の空間に存在する建築物などのオブジェクトを管理可能なデータベース管理システム（空間DB管理システム）が開発されている。このようなオブジェクトの検索は、ナビゲーション機能に特化した地図検索システム、都市計画や防災などの行政システム、ガスや水道などの施設管理を応用とする地理情報システムに必要とされる。そのため、空間DB管理システムは、地理情報システムのデータ管理ミドルウェアとして利用されている。

【0003】

オブジェクトは、その位置や形状を表現する図形要素と、その内容を表現する文字や数値等の属性要素から構成される。空間DB管理システムでは、大容量の二次記憶装置に格納された多量のオブジェクトの中から、図形要素を含む検索条件に合致したオブジェクトを高速に検索することが求められる。

【0004】

空間DB管理システムでは、オブジェクトの検索として、範囲検索やk最近傍検索を備えている。範囲検索では、任意の形状の図形を指定し、その図形に包含される、または交差するオブジェクトを検索する。地理情報システムでは、範囲検索は、地図を画面に描画する際に用いられる。この場合、画面の範囲に含まれるオブジェクトを検索することになる。ユーザが描画範囲を移動させる際、または描画範囲の拡大や縮小を行う際に、範囲検索の処理が頻発する。

【0005】

一方、k最近傍検索では、任意の地点を指定し、その地点から距離の近い上位k件のオブジェクトを検索する。k最近傍検索は、例えば、地図検索システムにおいて、観光先の目的地から距離の近い上位10件の飲食店を検索する場合に用いられる。また、地図情報システムにおいて、図形データの編集を行う際に、マウスポインタで最近接のオブジェクトを指定する場合に、最近傍検索が利用される。

【 0 0 0 6 】

また、空間 D B 管理システムは、地図を利用する組込機器のソフトウェア（組込ソフトウェア）の開発でも、利用されつつある。その代表例は、カーナビゲーション装置や P N D（Personal Navigation Device）などの車載情報端末や携帯情報端末のナビゲーションソフトウェアである。ナビゲーションソフトウェアの主な機能は、画面への地図表示、店舗や施設などの地図検索、現在地から目的地までの経路検索などである。これらの機能の内部処理は、オブジェクトの検索を基本処理とする。開発者が空間 D B 管理システムを用いると、機能ごとにオブジェクトの検索の処理を実装する必要がないため、ソフトウェアの開発工数を削減できる。

【 0 0 0 7 】

10

空間 D B 管理システムは、オブジェクトの検索を高速に処理するため、空間索引を作成する。空間 D B 管理システムは、空間索引を用いることで、検索対象となるオブジェクトを空間的に絞り込むことができる。その結果、検索処理時間の大部分を占める二次記憶装置と主記憶装置の間のディスクアクセス処理の回数を削減できる。このディスクアクセス処理は、二次記憶装置から主記憶装置にオブジェクトを読み込むときに発生する。

【 0 0 0 8 】

非特許文献 1 には、代表的な空間索引として、P M R 4 分木をデータ構造とする手法が開示されている。P M R 4 分木では、再帰的に空間領域を 4 分割した領域を Z オータの順位で符号化し、その領域の符号値をキー値とした 1 次元のツリー構造索引で管理する。

【 先行技術文献 】

20

【 非特許文献 】

【 0 0 0 9 】

【 非特許文献 1 】「Speeding up construction of PMR quadtree-based spatial indexes」VLDB Journal、Vol.11、Issue.2、pp.109-137（2002）

【 発明の概要 】

【 発明が解決しようとする課題 】

【 0 0 1 0 】

組込ソフトウェアなどのハードウェア資源の使用制約が厳しいプラットフォーム上での開発では、次の 2 つの制約を考慮する必要がある。

【 0 0 1 1 】

30

1 つ目の制約は、組込ソフトウェアのサイズの制約である。組込ソフトウェアのプログラム、データ、索引を事前に決められた二次記憶装置や主記憶装置の容量に合わせる必要がある。組込システムの開発では、ハードウェアの部品を短小軽薄にしたいというニーズが高いため、開発者は、組込システムのハードウェアの部品に小容量の二次記憶装置や主記憶装置を用いる。その結果、組込ソフトウェアのサイズの制約を考慮し、空間 D B のサイズを削減する必要がある。空間 D B は、管理プログラム、オブジェクト、空間索引から構成される。これらの中で、大きな割合を占める空間索引の格納に必要な容量の削減が求められる。

【 0 0 1 2 】

2 つ目の制約は、組込ソフトウェアの処理時間の制約である。組込システムでは、ある処理を実行してから、決められた時間内にその処理を終了させることが求められる。

40

範囲検索は、オブジェクトの検索の中で、最もよく利用される。そこで、容量削減後の空間索引を用いた場合でも、容量削減前の空間索引と同等に、高速に範囲検索を実現できる必要がある。

【 0 0 1 3 】

前記した非特許文献 1 で使用される P M R 4 分木で空間データを管理する手法では、P M R 4 分木のデータ構造に冗長な部分が存在するため、2 つの制約（小容量かつ高速）を十分に満たしているとはいえない。

例えば、P M R 4 分木のデータ構造は、B 木などの一次元のツリー構造索引となり、リーフノードに空間領域の情報とオブジェクトの情報を組とした要素を格納する。この空間

50

領域は、オブジェクトを内包または交差する領域となる。ここで、1つの空間領域に複数のオブジェクトが内包また交差する場合、同じ空間領域の情報をもつ要素が重複して、格納されることになる。

【0014】

そこで、本発明は、前記した問題を解決し、空間領域上に配置されたオブジェクトの検索処理について、小容量かつ高速で実施することを、主な目的とする。

【課題を解決するための手段】

【0015】

前記課題を解決するために、本発明は、多次元空間上にオブジェクトが配置される空間データを管理する空間データ管理装置であって、

前記空間データ管理装置は、空間データを格納する記憶部と、指定された検索範囲内に存在するオブジェクトを前記空間データから検索するオブジェクト検索部と、を有し、

前記記憶部には、各ノードからの子ノードの数を制限するバランス木として前記空間データを格納する領域索引ツリーが記憶され、

前記記憶された領域索引ツリーの各ノードには、前記空間データを1回以上等分割した空間領域の情報が所定数以下だけ格納され、その空間領域の情報は、空間領域の多次元空間上の位置を一意に示すスカラ値に写像した結果であるキー値が含まれ、

前記領域索引ツリーの子ノードを持つ非リーフノードには、その各子ノードのとりうるキー値の範囲に応じて、各子ノードへのリンクが対応づけられており、前記領域索引ツリーの子ノードを持たないリーフノードには、空間領域の情報から、その空間領域に外接矩形が交わるオブジェクトへのリンクが対応づけられており、

前記オブジェクト検索部が、

入力装置を介して前記検索範囲が指定されると、

前記領域索引ツリーのルートノードから順に、前記検索範囲の代表点をスカラ値に写像した検索符号値と、前記領域索引ツリーの各ノードの空間領域のキー値とを照合することにより、非リーフノード、リーフノード、オブジェクトへのリンクと辿ることで、前記検索範囲の代表点が含まれる空間領域と、その空間領域からリンクするオブジェクトとを抽出する手順について、一巡ごとに前記検索範囲から辿った空間領域を除外した領域の代表点を新たな検索符号値として更新することで繰り返し、

前記検索範囲から辿った空間領域を除外した領域が無くなると、前記領域索引ツリーの各ノードを辿る処理により抽出される各オブジェクトについて、オブジェクトの外接矩形と前記検索範囲とを照合し、少なくとも一部が交わるオブジェクトを、前記検索範囲に交わるオブジェクトとして出力することの特徴とする。

その他の手段は、後記する。

【発明の効果】

【0016】

本発明によれば、空間領域上に配置されたオブジェクトの検索処理について、小容量かつ高速で実施することができる。

【図面の簡単な説明】

【0017】

【図1】本発明の一実施形態に関する空間データ管理装置を示す構成図である。

【図2】本発明の一実施形態に関する領域索引ツリーを示す説明図である。図2(a)は、空間領域の展開図を示し、図2(b)は、その展開図を作成するためのツリー構造を示す。

【図3】本発明の一実施形態に関する局所的な地域における領域索引ツリーを示す説明図である。図3(a)は、空間領域の展開図を示し、図3(b)は、その展開図を作成するためのツリー構造を示す。

【図4】本発明の一実施形態に関するオブジェクト検索部が実行する領域索引ツリーの検索処理の概要を示す説明図である。図4(a)は、空間領域の展開図を示し、図4(b)は、その展開図を作成するためのツリー構造を示す。

10

20

30

40

50

【図 5】本発明の一実施形態に関するオブジェクト登録部が実行する領域索引ツリーの登録処理の概要を示す説明図である。図 5 (a) は、空間領域の展開図を示し、図 5 (b) は、その展開図を作成するためのツリー構造を示す。

【図 6】本発明の一実施形態に関するオブジェクト削除部が実行する領域索引ツリーの削除処理の概要を示す説明図である。図 6 (a) は、空間領域の展開図を示し、図 6 (b) は、その展開図を作成するためのツリー構造を示す。

【図 7】本発明の一実施形態に関するオブジェクト検索部が実行する領域索引ツリーの検索処理を示すフローチャートである。

【図 8】本発明の一実施形態に関する領域索引ツリーの検索処理における空間領域の検索処理を示すフローチャートである。

10

【図 9】本発明の一実施形態に関する領域索引ツリーの検索処理におけるオブジェクトの交差判定処理を示すフローチャートである。

【図 10】本発明の一実施形態に関する領域索引ツリーの検索処理における検索符号値の更新処理を示すフローチャートである。

【図 11】本発明の一実施形態に関するオブジェクト登録部が実行する領域索引ツリーの登録処理を示すフローチャートである。

【図 12】本発明の一実施形態に関するオブジェクト登録部が実行する領域索引ツリーの登録処理の詳細を示すフローチャートである。図 12 (a) が登録のメイン処理であり、図 12 (b) はそのメイン処理から呼び出されるサブルーチン処理である。

【図 13】本発明の一実施形態に関するオブジェクト削除部が実行する領域索引ツリーの削除処理を示すフローチャートである。図 13 (a) が削除のメイン処理であり、図 13 (b) はそのメイン処理から呼び出されるサブルーチン処理である。

20

【発明を実施するための形態】

【0018】

以下、本発明の一実施形態について、図面を参照して説明する。

【0019】

図 1 は、空間データ管理装置 1 を示す構成図である。なお、本実施形態の「空間データ」における「空間」とは、3 次元の立体空間だけに限定するのではなく、 n 次元 (n は 2 以上の自然数) のベクトル空間全般を指す。以下、本実施形態は、 $n = 2$ の平面ベクトル空間を例示して説明するが、本実施形態は $n = 2$ だけに限定するものではなく、 n 次元に拡張可能である。なお、 n 次元空間において、1 回の空間分割処理により、1 つの空間が (2 の n 乗) 個の空間に等分割される。

30

【0020】

空間データ管理装置 1 は、中央処理装置 9 1、主記憶装置 9 2、二次記憶装置 9 3、入力装置 9 4、および、出力装置 9 5 を含むコンピュータとして構成される。

中央処理装置 9 1 は、主記憶装置 9 2 に格納されたプログラムの処理を行い、各種処理を実行する。

主記憶装置 9 2 は、中央処理装置 9 1 で実行されるプログラムやデータを格納するメモリとして構成される。主記憶装置 9 2 には、処理部として、オブジェクト検索部 3 0、オブジェクト登録部 4 0、オブジェクト削除部 5 0、および、データ切替部 6 0 を構成するためのプログラムが記憶されている。

40

二次記憶装置 9 3 は、ハードディスクドライブやフラッシュメモリディスクなどの大容量の記憶装置として構成される。二次記憶装置 9 3 には、主記憶装置 9 2 の各処理部が扱う空間データとして、領域索引ツリー 2 1 と、領域内オブジェクトテーブル 2 2 と、オブジェクト詳細テーブル 2 3 とを、対応づけて空間データベースとして記憶する。

入力装置 9 4 は、オブジェクトの検索に必要な検索条件などを入力する装置である。例えば、車載情報端末では、ハードウェアキーボードの他に、タッチパネルによるソフトウェアキーボードなどで構成される。

出力装置 9 5 は、オブジェクトの検索結果を表示する装置であり、液晶ディスプレイなどにより構成される。

50

【 0 0 2 1 】

二次記憶装置 9 3 が格納するデータについて、説明する。

領域索引ツリー 2 1 には、空間索引の情報が、低データ量かつ高速検索が可能なツリー構造として格納される（詳細は図 2 参照）。

領域内オブジェクトテーブル 2 2 には、領域索引ツリー 2 1 が示す各領域について、オブジェクトが含まれているか否かの情報が格納される。

オブジェクト詳細テーブル 2 3 には、領域内オブジェクトテーブル 2 2 で示されたオブジェクトの詳細情報が格納される。詳細情報には、オブジェクトの実形状を表現する座標値のリスト、および、オブジェクトの属性情報が含まれる。

【 0 0 2 2 】

オブジェクト検索部 3 0 は、領域索引ツリー 2 1 を用いて、任意の範囲（検索範囲）に含まれるオブジェクトを範囲検索する。

範囲検索には、次の二種類が存在する。1 つは、検索範囲に内包するオブジェクトを検索する場合である。もう 1 つは、検索範囲に内包または交差するオブジェクトを検索する場合である。本実施形態のオブジェクト検索部 3 0 は、交差関係も含めた後者の範囲検索を対象とする。後者の範囲検索は、車載情報端末の例では、自車位置や目的地の周辺の地図を車載情報端末に表示させる処理に用いられる。なお、前者の範囲検索は後者の範囲検索の条件を厳しくしたもので、基本的には同様の処理で実現できる。

範囲検索を実行するため、オブジェクト検索部 3 0 は、検索初期設定部 3 1、交差判定部 3 2、検索起点設定部 3 3、および、絶対座標値変換部 3 4 を含めて構成される。

【 0 0 2 3 】

オブジェクト登録部 4 0 は、二次記憶装置 9 3 の空間データ（領域索引ツリー 2 1、領域内オブジェクトテーブル 2 2、オブジェクト詳細テーブル 2 3）に対して、新たにオブジェクトを登録（追加）する。

オブジェクト登録部 4 0 は、相対座標値変換部 4 1 と、領域分割部 4 2 とを含めて構成される。

相対座標値変換部 4 1 は、オブジェクトの外接矩形を絶対座標値から相対座標値に変換する。

領域分割部 4 2 は、オブジェクトの挿入で領域索引ツリー 2 1 のオブジェクトノードが溢れた場合に、空間領域を 4 分割する。

【 0 0 2 4 】

オブジェクト削除部 5 0 は、二次記憶装置 9 3 の空間データ（領域索引ツリー 2 1、領域内オブジェクトテーブル 2 2、オブジェクト詳細テーブル 2 3）に対して、既に登録されているオブジェクトを削除する。オブジェクト削除部 5 0 は、領域併合部 5 1 を含めて構成される。

領域併合部 5 1 は、オブジェクトの削除によって領域索引ツリー 2 1 のオブジェクトノードを併合できる場合に、4 つの空間領域を 1 つにまとめる。

【 0 0 2 5 】

データ切替部 6 0 は、オブジェクトの各処理（検索処理、登録処理、削除処理）の対象となる空間データ（領域索引ツリー 2 1、領域内オブジェクトテーブル 2 2、オブジェクト詳細テーブル 2 3）のセットを、切り替える。空間データのセットは、二次記憶装置 9 3 内に複数組格納されている。

データ切替部 6 0 は、例えば、各空間データのセットが扱う空間範囲（日本全国版、関東版、東京都版など）をあらかじめ記憶し、入力装置 9 4 を介して指定される検索条件が示す位置（例えば、東京駅）が内包される空間データのセットの内、もっとも詳細な地図縮尺に対応する空間データのセット（例えば、東京都版）を選択する。

【 0 0 2 6 】

このように、空間データのセットを複数個用意することによって、二次記憶装置の容量を消費するものの、範囲検索を高速化させることができる。なお、空間データのセット（詳細地図）には、例えば、ドライバが普段頻繁に運転する地域が設定される。

10

20

30

40

50

一方、空間データのセットを１つ（例えば、日本全国版）だけ用意してもよい。そのときには、データ切替部 60 を空間データ管理装置 1 から省略することができる。

【 0 0 2 7 】

図 2 (a) は、領域索引ツリー 2 1 が示す情報を、説明用に理解しやすいように展開した展開図である。この展開図では、横 (x) 1 6 区画 × 縦 (y) 1 6 区画の空間領域を、上方 (z 軸) から見下ろした形式で示しており、7 つのオブジェクト (A ~ G) が配置されている。

展開図内の数字（例えば、原点「 0 」や中央点「 1 9 2 」など）は、その数字の記載位置に対応する Z オーダの順位で符号化したモートン符号値を示す。なお、モートン符号値は、n 次元の空間領域上の位置を 1 次元のスカラー値に写像する写像値の一例である。本実施形態では、モートン符号値を例示したが、その代わりにヒルベルト曲線で符号化したヒルベルト符号値などの他の写像演算値を用いてもよい。

展開図内の灰色矩形は、各オブジェクトが外接する矩形を示しており、その内部にオブジェクト ID (A ~ F) が記述されている。

展開図内の点線は、空間領域を各軸の垂直 2 等分線で等面積に 4 分割することで生成される各空間領域の境界を示す。例えば、モートン符号値「 1 9 2 」を起点（左下点）とする右上の領域は、1 6 区画の空間領域を 1 分割した結果の大きめの (8 × 8 区画) 領域である。一方、モートン符号値「 6 0 」を起点（左下点）とする中央付近の領域は、1 6 区画の空間領域を 3 分割した結果の小さめの (2 × 2 区画) 領域である。

【 0 0 2 8 】

空間データ管理装置 1 は、各空間領域の位置を、その代表点（左下点）を変換したモートン符号値と分割回数（分割数）の組で表現する。空間領域を再帰的に等面積で 4 分割するため、モートン符号値と分割数の組で空間領域の位置と形状を一意に特定できる。空間領域の代表点は、領域内の x 座標と y 座標の最小値を組とする座標値とする。

【 0 0 2 9 】

以下、座標値 (7 , 1 0) をモートン符号値「 1 5 7 」に変換する計算方法を示す。

（手順 1）座標値として、(x , y) = (7 , 1 0) が入力される。

（手順 2）入力された 1 0 進数の座標値 (7 , 1 0) を、2 進数の座標値 (0 1 1 1 , 1 0 1 0) に変換する。

（手順 3）2 進数の座標値 (0 1 1 1 , 1 0 1 0) を、y 座標と x 座標の順番で、2 進数の先頭ビットから順に交互に並べ、「 1 0 0 1 1 1 0 1 」とする。

「 1 0 0 1 1 1 0 1 」の先頭の「 1 」は、y 座標「 1 0 1 0 」の先頭の「 1 」である。

「 1 0 0 1 1 1 0 1 」の 2 桁目の「 0 」は、x 座標「 0 1 1 1 」の先頭の「 0 」である。

。

「 1 0 0 1 1 1 0 1 」の 3 桁目の「 0 」は、y 座標「 1 0 1 0 」の 2 桁目の「 0 」である。

（手順 4）「 1 0 0 1 1 1 0 1 」を 1 0 進数に変換すると、モートン符号値「 1 5 7 」となる。

【 0 0 3 0 】

モートン符号値の性質より、空間領域内の任意の矩形の左下座標に対応したモートン符号値は、必ずその矩形の右上座標に対応したモートン符号値より小さくなる。また、矩形内部の座標に対応したモートン符号値は、矩形の左下座標と右上座標のモートン符号値の間に含まれることになる。多次元の範囲検索にモートン符号値のこれらの性質を利用すると、検索条件となる多次元の範囲を 1 次元のスカラー値の範囲に簡単化できる。さらに、モートン符号値は、空間的に近い座標値の間で、近い符号値を割り当てることができるため、1 次元のスカラー値の範囲を狭め、範囲検索を高速化できる。

【 0 0 3 1 】

前記の例では、2 次元平面 (x , y) 上でのモートン符号値の計算方法を示したが、モートン符号値は、2 次元に限定されることはなく、3 次元以上の n 次元にも拡張することができる。n 次元のモートン符号値の計算は、前記した（手順 3）を、以下の（手順 3 b

10

20

30

40

50

）に置き換えることで、実現できる。

（手順 3 b）2 進数で表記した n 次元の座標値を、以下の順序で交互に並べる。

（ n 次元座標値の先頭ビット） （ $n - 1$ 次元座標値の先頭ビット） （略） （1 次元座標値の先頭ビット） （ n 次元座標値の 2 桁目のビット） （ $n - 1$ 次元座標値の 2 桁目のビット） （略） （1 次元座標値の 2 桁目のビット） （略） （1 次元座標値の最終ビット）

【 0 0 3 2 】

図 2（b）は、図 2（a）が示す領域索引ツリー 2 1 の情報を、ツリー構造として示した構成図である。領域索引ツリー 2 1 は、各ノードと、ノード間を接続するリンクとがツリー構造になっている。領域索引ツリー 2 1 は、バランス木（B 木、B + 木など）のツリー構造であり、非リーフノード、リーフノード、オブジェクトノードから構成される。これらのノードは、ページという単位で 1 対 1 の関係で二次記憶装置に格納される。ページは、データベース管理システムにおいて、二次記憶装置と主記憶装置の間の最小の入出力単位である。

以下の説明では、バランス木の一例として、B + 木を用いる例を説明する。B + 木は、検索と更新の処理を高速に行えるという点で、データベース管理システムで最もよく用いられている索引構造である。B + 木では、非リーフノードにはキー値と子ノードへのポインタ、リーフノードにはキー値とレコードポインタが含まれる。そして、隣接するリーフノードの間を片方向のリンクで連結する。

【 0 0 3 3 】

領域索引ツリー 2 1 のリーフノード、非リーフノードには、キー値であるモートン符号値（図中では数値で示す）と、そのモートン符号値が示す左下座標からの空間領域を作成するための分割数（図中では括弧内の数値で示す）と、の組（図中では各ノードの灰色部で示す）が、最大 3 組分格納できる。例えば、図 2（a）の左上の空間領域は、「モートン符号値 = 1 2 8 , 分割数 = 1」の組として表現される。

このモートン符号値と分割数との組が図 2（a）の 1 つ分の空間領域を示すので、以下の説明では、モートン符号値と分割数との組を「ノード内の空間領域情報」と表記する。

このように、空間領域を等面積で 4 分割することで生成される空間領域を、モートン符号値と分割数との組で一意に表現する。モートン符号値は、一次元のスカラ値なので、空間領域の次数（2 次元平面、3 次元空間）などに依存せず大小関係を定義でき、B + 木のノードのキー値（一次元のキー値）として機能する。

【 0 0 3 4 】

領域索引ツリー 2 1 の非リーフ（幹）ノードは、最大 3 件の空間領域情報を格納する。非リーフノードは、非リーフノードまたはリーフノードを子ノードとしてリンクする。非リーフノード内の空間領域情報は、降順または昇順に配置されており、空間領域情報間の白色部には、子ノードへのポインタが矢印で表記される。子ノードへのポインタは、子ノードに対応したページを格納しているディスク上の物理的アドレスを示す。

なお、空間領域情報の最大個数を 3 件とするのは、あくまで一例である。空間領域情報の最大個数は、ページサイズ、空間領域情報のサイズ、子ノードへのポインタのサイズなどで決定される。また、B + 木のデータ構造上、空間領域情報の最大個数が「 i 」個の場合、子ノードへのポインタは「 $i + 1$ 」個となる。

親ノードの第 1 空間領域情報と第 2 空間領域情報との間に設定されるポインタは、そのポインタが示す子ノードを指し示す。このとき、子ノードの各空間領域情報のモートン符号値は、第 1 空間領域情報のモートン符号値から第 2 空間領域情報のモートン符号値までの範囲に収まる。

【 0 0 3 5 】

領域索引ツリー 2 1 のリーフ（葉）ノードは、最大 3 件の空間領域情報を格納する。リーフノードは、オブジェクトノードにリンクする。リーフノードの空間領域情報は、降順または昇順に配置されている。リーフノードの各空間領域情報（例えば、モートン符号値「3 2」）からは、その空間領域情報が示す空間領域に重なるオブジェクトを示すオブジ

10

20

30

40

50

エクトノードへのリンク（例えば、オブジェクト「A」）が存在する。

なお、空間領域情報の最大個数を3件とするのは、あくまで一例であり、最大個数を「i」個とすると、「i」個のオブジェクトノードへのポインタが格納される。

【0036】

さらに、同じリーフノード内に格納される複数の空間領域情報は、互いに空間領域情報のモートン符号値が重複しないように、リーフノードを構成する。

これにより、領域索引ツリー21に必要なディスク容量を、以下の比較例よりも削減することができる。その比較例とは、キー値とオブジェクトの情報の組を領域索引のリーフノードの要素として、PMR4分木を構成する例である。この比較例では、1つの空間領域が複数のオブジェクトを内包するため、同じキー値の要素がリーフノードに複数格納され、冗長になる。

10

【0037】

領域索引ツリー21のオブジェクトノードは、リンクされるリーフノードが示す空間領域の情報1件と、その空間領域に重なる（一部接触または全部内包）複数のオブジェクトの情報（この場合、最大3件）を格納する。オブジェクトの情報は、後述する領域内オブジェクトテーブル22内の空間領域（モートン符号値と分割数）、オブジェクト（IDと領域内位置）を含む。

【0038】

【表1】

22 領域内オブジェクトテーブル

20

空間領域		オブジェクト	
モートン符号値	分割数	ID	領域内位置 (左下座標～右上座標)
16	2	E	(0, 2)～(2, 2)
32	2	A	(3, 0)～(3, 3)
48	3	A	(0, 0)～(0, 1)
52	3	F	(0, 0)～(1, 0)
		G	(0, 1)～(1, 1)
56	3	A	(0, 0)～(0, 1)
60	3	D	(0, 1)～(1, 1)
64	1	G	(0, 5)～(1, 5)
128	1	C	(2, 3)～(4, 4)
		D	(6, 0)～(7, 1)
192	1	B	(3, 1)～(6, 3)

30

【0039】

表1は、領域内オブジェクトテーブル22を示す。領域内オブジェクトテーブル22は、空間領域情報（モートン符号値および分割数の組）と、オブジェクト情報（オブジェクトIDおよびオブジェクトの領域内位置の組）とを対応づけて管理する。1つの空間領域の情報と、その空間領域に対応する複数のオブジェクトの情報は、領域索引ツリーのオブジェクトノードに含まれる。1つのオブジェクトノードに1ページを割り当てるため、同じ空間領域に重なるオブジェクトの情報は1ページにまとめて格納される。これにより、同じ領域に重なるオブジェクトの情報を1回のディスクアクセスで取得できるため、範囲検索の高速化に寄与する。

40

領域内オブジェクトテーブル22の空間領域情報は、領域索引ツリー21の非リーフノードおよびリーフノード内の空間領域情報と同じである。

【0040】

領域内オブジェクトテーブル22のオブジェクトIDは、領域索引ツリー21のオブジェクトノードのオブジェクトIDと同じである。オブジェクトIDとは、オブジェクトを一意に特定するための情報であり、たとえば、オブジェクトのレコードポインタである。

50

オブジェクトのレコードポインタは、テーブル内のレコードを格納したディスク上の物理的アドレスを示す。

【 0 0 4 1 】

領域内オブジェクトテーブル 2 2 のオブジェクトの領域内位置は、オブジェクト ID が示すオブジェクトの外接矩形（左下座標～右上座標）の相対座標値を示す。

例えば、空間領域（モートン符号値 = 1 6、分割数 = 2）に含まれるオブジェクト（オブジェクト ID = E）は、外接位置が「(0 , 2) ~ (2 , 2)」である。

そして、図 2 (a) を参照すると、オブジェクト「E」は、左下座標 (4 , 0) ~ 右上座標 (7 , 3) の空間領域に内包されている。

オブジェクト「E」の左下座標は、内包される空間領域の左下座標 (4 , 0) に、領域内オブジェクトテーブル 2 2 の左下座標 (0 , 2) を加算することで、(4 , 2) と計算できる。

10

オブジェクト「E」の右上座標は、内包される空間領域の左下座標 (4 , 0) に、領域内オブジェクトテーブル 2 2 の右上座標 (2 , 2) を加算することで、(6 , 2) と計算できる。

【 0 0 4 2 】

ここで、各パラメータの表現に必要な最低限のビット数を事前に定めておくことにより、ディスク容量を削減できる。

例えば、モートン符号値の表現に必要な最低限のビット数は、図 2 の例では、空間領域の範囲が 16×16 なので、1 6 (2 の 4 乗) が 2 つ分で 8 ビットとなる。つまり、地図の範囲を示す空間領域の大きさをもとに、モートン符号値を格納する変数の容量を特定することができる。

20

一方、領域内位置の座標の表現に必要な最低限のビット数は、例えば、空間領域（モートン符号値 = 1 2 8、分割数 = 1）の場合、 8×8 の領域であるため、x 座標と y 座標とでそれぞれ 3 ビットであるので合計 6 ビットとなる。オブジェクトの領域内位置を、内包される空間領域の代表点（左下座標）からの差分形式（相対座標）で表現することで、PMR 4 分木が必要とするデータ容量と比べ、領域内位置の格納に必要なディスク容量を削減できる。このディスク容量の削減効果は、適用するハードウェア部品やアプリケーションにより、二次記憶装置が非常に小さい組込機器への適用に、特に有効である。

このように、扱う領域の大きさに応じて、変数のビット数を規定することができる。なお、ビット数単位とする代わりに、バイト数単位としてもよい。

30

【 0 0 4 3 】

【表 2】

2 2 領域内オブジェクトテーブル (種別 I D を付加)

空間領域		オブジェクト		
モートン 符号値	分割数	I D	領域内位置 (左下座標～右上座標)	種別 I D
1 6	2	E	(0, 2)～(2, 2)	1 (ガソリンスタンド)
3 2	2	A	(3, 0)～(3, 3)	2 (飲食店)
4 8	3	A	(0, 0)～(0, 1)	2 (飲食店)
5 2	3	F	(0, 0)～(1, 0)	0 (銀行)
		G	(0, 1)～(1, 1)	2 (飲食店)
5 6	3	A	(0, 0)～(0, 1)	2 (飲食店)
6 0	3	D	(0, 1)～(1, 1)	1 (ガソリンスタンド)
6 4	1	G	(0, 5)～(1, 5)	2 (飲食店)
		C	(2, 3)～(4, 4)	3 (学校)
1 2 8	1	D	(6, 0)～(7, 1)	1 (ガソリンスタンド)
		B	(3, 1)～(6, 3)	0 (銀行)

10

【0 0 4 4】

20

表 2 は、領域内オブジェクトテーブル 2 2 を示す。表 1 と比較すると、オブジェクトの情報に「種別 I D」列が追加されている。

種別 I D は、オブジェクトの種類を示す。例えば、種別 I D には、銀行「0」、ガソリンスタンド「1」、飲食店「2」、学校「3」などが含まれる。これらの種別 I D は、入力装置 9 4 を介して入力されるユーザからの検索条件において、指定される。例えば、ユーザは、自車から半径 5 0 0 m 以内の施設の内、「ガソリンスタンド」だけを選択する旨の検索条件を入力する。

種別 I D の表現に必要な最低限のビット数は、種別 I D のとりうる値の範囲を元に、設定される。例えば、種別 I D が 1 0 0 通り存在するときには、2 の 8 乗 = 1 2 8 通りの値を格納できる変数に種別 I D を格納するため、8 ビットが最低限のビット数となる。

30

【0 0 4 5】

【表 3】

2 3 オブジェクト詳細テーブル

オブジェクト I D	種別 I D	実形状
A	2 (飲食店)	長方形
B	0 (銀行)	楕円形
C	3 (学校)	コの字型
D	1 (ガソリンスタンド)	長方形
E	1 (ガソリンスタンド)	L 字型
F	0 (銀行)	円形
G	2 (飲食店)	長方形

40

【0 0 4 6】

表 3 は、オブジェクト詳細テーブル 2 3 を示す。オブジェクト詳細テーブル 2 3 は、個々のオブジェクトに関する詳細な情報として、オブジェクト I D と、種別 I D と、実形状と、を対応づけて格納する。

オブジェクト詳細テーブル 2 3 の種別 I D は、領域内オブジェクトテーブル 2 2 の種別 I D と同じ意味である。

50

オブジェクト詳細テーブル 23 の実形状はオブジェクトの実際の形状を示し、例えば、オブジェクトの形状を座標の列で表現できる。オブジェクトの実形状は、オブジェクトの検索処理において、高精度な検索をするために用いられる。例えば、オブジェクトの外接矩形が検索範囲内に収まっていたとしても、オブジェクトが円であるときには、外接矩形の角が少し検索範囲内に重なるだけでは、オブジェクトの円は検索範囲内に収まっていないことになる。

【 0 0 4 7 】

【表 4】

2.2 領域内オブジェクトテーブル

空間領域		オブジェクト	
モートン符号値	分割数	ID	領域内位置 (左下座標～右上座標)
0	1	A	(0, 1)～(1, 3)
		G	(3, 2)～(3, 2)
		F	(3, 1)～(3, 1)
1 6	1	F	(0, 1)～(0, 1)
		G	(0, 2)～(2, 2)
3 2	1	A	(0, 0)～(1, 0)
		D	(3, 0)～(3, 2)
4 8	1	D	(0, 0)～(0, 2)

【 0 0 4 8 】

以上説明した空間データ（領域索引ツリー 21、領域内オブジェクトテーブル 22、オブジェクト詳細テーブル 23）のセットは、二次記憶装置 93 内に 1 組以上格納されている。データ切替部 60 は、オブジェクトの各処理（検索処理、登録処理、削除処理）の対象となる空間データのセットを切り替える。

図 3 は、図 2 の左下座標値（3, 3）から右上座標値（10, 10）までの矩形領域を局所的な地域とした、領域索引ツリー 21 とその展開図を示す。

表 4 は、図 3 の領域索引ツリー 21 とセットになる領域内オブジェクトテーブル 22 の一例を示す。このように、空間データのセットを局所化することにより、各データ構造の扱う範囲が狭まり、検索対象のデータ量が少なくなる。その結果、範囲検索で二次記憶装置から主記憶装置に読み込むノード（ページ）が少なくなり、検索処理時間を短縮できる。

【 0 0 4 9 】

図 4 は、オブジェクト検索部 30 が実行する、検索範囲を、太い点線で囲む矩形の左下座標（5, 2）～右上座標（7, 5）としたときの、図 2 の領域索引ツリー 21 を検索する様子を示す。以下に、この場合の手順について、図 4（a）および図 4（b）を併せて参照しつつ、説明する。

【 0 0 5 0 】

まず、図 4（b）における 1 回目の検索処理（一番左の点線矢印）について、説明する。検索初期設定部 31 は、検索範囲の左下座標（5, 2）をモートン符号値「28」に変換し、検索符号値 M に設定する。領域索引ツリーを探索する際、検索符号値 M に対応した座標を含む空間領域を辿る。モートン符号値の性質から、検索符号値 M 以下で最大のモートン符号値を含む部分木を辿ることになる。この場合、領域索引ツリー 21 のルートノードを参照すると、モートン符号値「60」の左の子ノードへのリンクを辿る。これは、このリンクは、検索符号値 M「28」以下で最大のモートン符号値をもつ部分木を指し示すためである。

ルートノードの子ノードでは、同様にモートン符号値「32」の左のポインタが指し示す孫ノードを辿る。さらに、この孫ノードでは、検索符号値 M「28」は空間領域情報の

モートン符号値「16」を代表点とする空間領域に含まれるため（空間領域の右上座標のモートン符号値は31であるため）、そのノードからのリンク先であるオブジェクトノード「E」を概略判定リストに入れる。

【0051】

検索起点設定部33は、領域索引ツリー21をリーフノード（およびその下のオブジェクトノード）まで辿った後、検索符号値Mを更新する。今回の検索符号値Mは、今回の検索符号値Mに対応する検索範囲から、今回の検索符号値Mから検索された空間領域を除外（領域として切り取る）した後の、残りの検索範囲の座標に対応したモートン符号値の中で最小値になる。

例えば、今回の検索範囲（5, 2）～（7, 5）に対して、検索された空間領域の範囲（4, 0）～（7, 3）を除外すると、残りの検索範囲は（5, 4）～（7, 5）となる。よって、検索起点設定部33は、座標（5, 4）のモートン符号値「49」を次の検索対象Mとする。

【0052】

次に、図4（b）における2回目の検索処理（中央の点線矢印）について、説明する。領域索引ツリー21のルートノードを参照すると、検索符号値M「49」以下で最大のモートン符号値をもつ部分木を指し示すモートン符号値「60」の左の子ノードへのリンクを辿る。その子ノードでは、空間領域情報のモートン符号値「32」以上で、「52」よりも小さい左から2つめの子ノードへのリンクを辿る。その子ノード（ルートノードの孫ノード）では、検索符号値M「49」は空間領域情報のモートン符号値「48」を代表点とする空間領域に含まれるため、そのノードからのリンク先であるオブジェクトノード「A」を概略判定リストに入れる。

今回の検索範囲（5, 4）～（7, 5）に対して、検索された空間領域の範囲（4, 4）～（5, 5）を除外すると、残りの検索範囲は（6, 4）～（7, 5）となる。よって、検索起点設定部33は、座標（6, 4）のモートン符号値「52」を次の検索対象Mとする。

【0053】

そして、図4（b）における3回目の検索処理（一番右の点線矢印）について、説明する。領域索引ツリー21のルートノードを参照すると、検索符号値M「52」はルートノードの空間領域情報のモートン符号値「60」よりも小さいので、左下の子ノードへのリンクを辿る。その子ノードでは、検索符号値M「52」は空間領域情報のモートン符号値「52」以上なので、さらに左から3つめの子ノードへのリンクを辿る。その子ノード（ルートノードの孫ノード）では、検索符号値M「52」は空間領域情報のモートン符号値「52」を代表点とする空間領域に含まれるため、そのノードからのリンク先であるオブジェクトノード「F, G」を概略判定リストに入れる。

今回の検索範囲（6, 4）～（7, 5）に対して、検索された空間領域の範囲（6, 4）～（7, 5）を除外すると、残りの検索範囲が無くなる。よって、検索符号値Mは更新せず、領域索引ツリー21を対象とした検索を終了する。

【0054】

以上の結果、オブジェクト「E, A, F, G」が、領域索引ツリー21からリンクを辿ることにより検索される。この後の詳細判定（詳細は後記する交差判定部32の処理を参照）により、オブジェクト「A」は、検索範囲外として除外され、オブジェクト「E, F, G」が、図4（a）の太い破線の枠で囲んだ検索範囲（5, 2）～（7, 5）に適合する検索結果として、出力装置95から出力される。

【0055】

図5は、オブジェクト登録部40が実行する、図2の領域索引ツリー21の状態から、オブジェクトH, Iを登録する旨を示す説明図である。

図5（a）において、オブジェクトH（6, 11）～（8, 12）と、オブジェクトI（2, 15）～（6, 15）と、を順に空間領域（0, 8）～（7, 15）付近に挿入すると、その空間領域には4つのオブジェクトC, D, H, Iが含まれてしまい、許容量の

10

20

30

40

50

3つを超過する。

そこで、領域分割部42は、許容量が超過した1つの空間領域(0, 8) ~ (7, 15)を、4分割し、4つの空間領域(モートン符号値 = 128、144、160、176)を作成する。これにより、4つの空間領域にそれぞれ含まれるオブジェクトの個数を、許容量の3つ以内にすることができる。

図5(b)に示す領域索引ツリー21は、領域分割部42による領域分割処理に伴い、図2における1つの空間領域(モートン符号値 = 128、分割数 = 1)が、分割数 = 2とする4つの空間領域(モートン符号値 = 128、144、160、176)へと分割されている。この分割に伴い、領域索引ツリー21のモートン符号値 = 128の空間領域情報を含むリーフノードも、許容量である3つの空間領域情報を超過したため、領域索引ツリー21のリーフノードも2つに分割されている。

さらに、領域索引ツリー21のリーフノードの更新に伴い、そのリーフノードのリンク先となるオブジェクトノードも、併せて更新される。

【0056】

図6は、オブジェクト削除部50が実行する、図2の領域索引ツリー21の状態から、オブジェクトAを削除する旨を示す説明図である。

図6(a)において、オブジェクトA(3, 4) ~ (4, 7)を削除することにより、そのオブジェクトAが位置していた近辺の4つ組の空間領域(モートン符号値 = 48、52、56、60)は、それぞれオブジェクトの許容量(3つ)に余裕ができる。

そこで、領域併合部51は、4つ組の空間領域を、1つの空間領域(モートン符号値 = 48、分割数 = 2)に併合する。これにより、領域索引ツリー21のノード数を削減し、領域索引ツリーの格納に必要な二次記憶装置の容量を抑制できる。

図6(b)に示す領域索引ツリー21は、領域併合部51による領域併合処理に伴い、図2における4つの空間領域(モートン符号値 = 48、52、56、60)が、1つの空間領域(モートン符号値 = 48)へと併合されている。この併合に伴い、領域索引ツリー21の併合されたリーフノードのリンク先となるオブジェクトノードも、併合された空間領域内のオブジェクトD、G、Fを示すオブジェクトノードへと、併せて更新される。

【0057】

図7は、図4で示した、範囲指定された検索範囲内に存在するオブジェクトの検索処理を示すフローチャートである。このフローチャートは、オブジェクト検索部30によって実行される。

【0058】

S101において、検索初期設定部31は、入力装置94を介して、検索条件となる検索範囲の入力を受け付ける。検索範囲の形状は、円、矩形、多角形など任意である。例えば、「現在値を中心に半径500m以内に存在するガソリンスタンドを検索」が選択されたとき、以下の処理を行う。

- ・現在値を中心とした半径500mの円に外接する矩形を求め、その矩形の左下座標値をモートン符号値「M_min」に変換し、その矩形の右上座標値をモートン符号値「M_max」に変換する。

- ・モートン符号値「M_min」を、検索符号値Mの初期値にする。

【0059】

S102において、オブジェクト検索部30は、検索符号値Mを起点とする矩形領域と少なくとも一部が交わる空間領域を、領域索引ツリー21から検索する(その詳細は、図8を参照して後記する)。

S103において、交差判定部32は、S102で検索された空間領域内のオブジェクトの形状(領域内オブジェクトテーブル22、オブジェクト詳細テーブル23から取得)と、S101で指定された検索範囲の形状とを比較することで、検索範囲内に少なくとも一部が交わるオブジェクトを検索する(その詳細は、図9を参照して後記する)。

【0060】

S104において、検索起点設定部33は、S101で初期化されて、このS104で

10

20

30

40

50

更新される検索符号値Mについて、S 1 0 2で検索された空間領域を除外した後の検索範囲を示す検索符号値Mに更新する（その詳細は、図 1 0を参照して後記する）。

S 1 0 5において、オブジェクト検索部 3 0は、S 1 0 4で更新された検索符号値Mが、S 1 0 1で入力されたモートン符号値「M_max」より大きいかなかを判定する。判定条件を満たすときには処理を終了し、判定条件を満たさないときにはS 1 0 2へ戻る。

【 0 0 6 1 】

図 8は、オブジェクト検索部 3 0が実行する、S 1 0 2の処理の詳細を示すフローチャートである。本フローチャートでは、領域索引ツリーを用いて、検索符号値Mに対応した座標を含む空間領域を探す。このとき、検索符号値M以下で最大のモートン符号値をもつ空間領域を探すことになる。

10

【 0 0 6 2 】

S 2 0 1において、領域索引ツリー 2 1のルートノードを検索対象ノードNに設定する。

S 2 0 2において、検索対象ノードNが非リーフノードであるかなかを判定する。判定条件を満たすときにはS 2 0 3へ、判定条件を満たさないときにはS 2 1 1へ、それぞれ進む。

【 0 0 6 3 】

S 2 0 3において、非リーフノードのポインタを1件取得する。このとき、非リーフノードのポインタは先頭から順に取得される。

S 2 0 4において、ポインタの指すノードに対応した空間領域のモートン符号値の範囲を算出する。モートン符号値の範囲は、当該非リーフノードに含まれるキー値（モートン符号値、分割数）から算出できる。

20

S 2 0 5において、検索符号値MがS 2 0 4で算出したモートン符号値の範囲に含まれるかなかを判定する。S 2 0 5は、検索符号値以下で最大のモートン符号値を探す処理になる。判定条件を満たすときにはS 2 0 6へ、判定条件を満たさないときにはS 2 0 3へ、それぞれ進む。

S 2 0 6において、ポインタの指すノードを検索対象ノードNに設定し、S 2 0 2に戻る。

【 0 0 6 4 】

S 2 1 1において、リーフノードの要素を1件取得する。このとき、リーフノードの要素を先頭から順に取得する。

30

S 2 1 2において、S 2 1 1で取得した要素に対応する空間領域のモートン符号値の範囲を算出する。空間領域のモートン符号値の範囲は、当該リーフノードのキー値（モートン符号値、分割数）から算出できる。

S 2 1 3において、検索符号値MがS 2 1 2で算出したモートン符号値の範囲に含まれるかなかを判定する。判定条件を満たすときにはS 2 1 4へ、判定条件を満たさないときにはS 2 1 1へ、それぞれ進む。

S 2 1 4において、当該リーフノードの要素を検索された空間領域として保持し、処理をS 1 0 3に戻す。

【 0 0 6 5 】

40

図 9は、主に交差判定部 3 2（S 3 0 4は絶対座標値変換部 3 4）が実行する、S 1 0 3の処理の詳細を示すフローチャートである。

【 0 0 6 6 】

S 3 0 1において、S 1 0 2で保持された空間領域の要素の指すオブジェクトノードを取得する。

S 3 0 2において、オブジェクトノードの要素をすべて取得したかなかを判定する。判定条件を満たすときにはS 1 0 4に処理を戻し、判定条件を満たさないときにはS 3 0 3へ進む。

S 3 0 3において、オブジェクトノードの要素を1件取得する。このとき、オブジェクトノードの要素を先頭から順に取得する。以下、取得したオブジェクトノードの要素を、

50

「選択オブジェクト」と呼ぶ。

【0067】

S303bにおいて、領域内オブジェクトテーブル22の「種別ID」を参照し、選択オブジェクトの種別が、検索条件の種別と一致するか否かを判定する。判定条件を満たすときにはS304へ、判定条件を満たさないときにはS302へ、それぞれ進む。

なお、このS303bの判定処理は、領域内オブジェクトテーブル22に種別IDが含まれている場合（表2参照）に実行されるが、領域内オブジェクトテーブル22に種別IDが含まれていない場合（表1参照）は、省略可能である。

【0068】

S304において、絶対座標値変換部34は、以下の手順により、選択オブジェクトの
10 外接矩形の絶対座標値の変換を行う。

- ・領域内オブジェクトテーブル22のモートン符号値を、S102で求めた空間領域の左下座標値（a, b）に変換する。

- ・選択オブジェクトの領域内位置の左下座標値（x'_min, y'_min）に、空間領域の左下座標値（a, b）を加算して、選択オブジェクトの左下座標の絶対座標値（x_min, y_min）を求める（例えば、 $x_{min} = x'_{min} + a$ 、 $y_{min} = y'_{min} + b$ ）。

- ・選択オブジェクトの領域内位置の右上座標値（x'_max, y'_max）に、空間領域の左下座標値（a, b）を加算して、選択オブジェクトの右上座標の絶対座標値（x_max, y_max）を求める（例えば、 $x_{max} = x'_{max} + a$ 、 $y_{max} = y'_{max} + b$ ）。

【0069】

S305において、S304で求めた選択オブジェクトの外接矩形（x_min, y_min）～（x_max, y_max）が、S102で求めた空間領域と交差するか否かを判定する。判定条件を満たすときにはS306へ、判定条件を満たさないときにはS302へ、それぞれ進む。
20

なお、S305の交差判定処理は、2つの矩形（第1矩形と第2矩形）の間に重複箇所が存在するか否かの判定処理であり、例えば、第1矩形と第2矩形との間の位置関係を示す判定式をあらかじめ用意し、その判定式に代入することにより、実現される。判定式の一例として、「（第1矩形の右端のx座標）<（第2矩形の左端のx座標）」なら、第1矩形と第2矩形とは重複しない旨の判定式が例示される。

【0070】

S306において、S304で求めた選択オブジェクトの実形状をオブジェクト詳細テーブル23から読み取り、S101で入力された空間領域の実形状と交差するか否かを判定する。判定条件を満たすときにはS307へ、判定条件を満たさないときにはS302へ、それぞれ進む。
30

なお、S306の交差判定処理は、オブジェクト詳細テーブル23の「実形状」データをもとに行われる詳細な交差判定である。この交差判定処理は、例えば、選択オブジェクトの実形状上の絶対位置を順に網羅するように走査し、選択オブジェクトの各走査点のうち、空間領域内に存在する選択オブジェクトの走査点が1つ以上発見されたときには、交差すると判定する方法がある。

ここで、S306の交差判定処理は、実形状という大きいデータ量を対象とする判定処理であるので、計算量が大きくなってしまふ。そこで、S306の交差判定処理を実行するか否かをオプションとしてユーザに選択させ、ユーザが選択しないときには、S306の交差判定処理を省略し、S305を満たすときにはS307に進むようにしてもよい。
40

【0071】

S307において、空間領域と交差すると判断された選択オブジェクトを検索結果として出力装置95を介して出力し、処理をS302に戻す。

【0072】

図10は、検索起点設定部33が実行する、S104の検索符号値Mの更新処理の詳細を示すフローチャートである。本フローチャートでは、検索範囲の矩形が含むモートン符号値の中で、S102で検索された空間領域を除外した検索範囲のうちの最小のモートン
50

符号値を求める。

【 0 0 7 3 】

S 4 0 1 において、検索符号値 M を座標値 (a , b) に変換する。

S 4 0 2 において、S 1 0 2 で検索された空間領域の右上座標 (x_max , y_max) に設定する。

S 4 0 3 において、S 1 0 2 で検索された空間領域の右辺より x 軸方向に 1 だけ大きい y 軸に平行な線が検索範囲の矩形に交差するか否かを判定する。判定条件を満たすときには S 4 0 4 へ、判定条件を満たさないときには S 4 0 5 へ、それぞれ進む。

S 4 0 4 において、S 1 0 2 で検索された空間領域の近接点 (x_max + 1 , b) のモートン符号値を、検索符号値リストに挿入する。なお、検索符号値リストは、例えば、モートン符号値を優先度とした優先度付きキューとして実装される。

10

【 0 0 7 4 】

S 4 0 5 において、S 1 0 2 で検索された空間領域の上辺より y 軸方向に 1 だけ大きい x 軸に平行な線が検索範囲の矩形に交差するか否かを判定する。判定条件を満たすときには S 4 0 6 へ、判定条件を満たさないときには S 4 0 7 へ、それぞれ進む。

S 4 0 6 において、S 1 0 2 で検索された空間領域の近接点 (a , y_max + 1) のモートン符号値を、検索符号値リストに挿入する。

S 4 0 7 において、S 4 0 4 および S 4 0 6 で挿入された検索符号値リストのモートン符号値のうちの最小のモートン符号値を、検索符号値 M に設定する。

20

【 0 0 7 5 】

図 1 1 は、オブジェクト登録部 4 0 が実行する、オブジェクトを空間データに登録する処理を示すフローチャートである。

【 0 0 7 6 】

S 5 0 1 において、オブジェクト登録部 4 0 は、登録対象のオブジェクトの詳細情報を取得する。

詳細情報とは、例えば、オブジェクト詳細テーブル 2 3 に登録される情報 (オブジェクトの地図上の位置、種別 ID、実形状) である。さらに、詳細情報として、カーナビゲーションの施設情報として提供される任意の情報 (住所、電話番号など) を付加してもよい。

【 0 0 7 7 】

30

S 5 0 2 において、オブジェクト登録部 4 0 は、S 5 0 1 で入力されたオブジェクトの詳細情報を、空間データ (オブジェクト詳細テーブル 2 3) に登録する。なお、登録対象のオブジェクトが既にオブジェクト詳細テーブル 2 3 に登録済みであるときには、2 重登録を避けるため、S 5 0 2 を省略してもよいし、S 5 0 1 で入力された詳細情報に変更があれば、登録済みのオブジェクト詳細テーブル 2 3 のレコードを、変更後の値に書き換えてもよい。

【 0 0 7 8 】

S 5 0 3 において、オブジェクト登録部 4 0 は、S 5 0 2 でのオブジェクト詳細テーブル 2 3 への登録データをもとに、関連する他の空間データ (領域索引ツリー 2 1、領域内オブジェクトテーブル 2 2) への登録データを作成する。

40

例えば、オブジェクト登録部 4 0 は、オブジェクト詳細テーブル 2 3 に新たに登録されたオブジェクトを、領域内オブジェクトテーブル 2 2 に登録されている各空間領域に沿って分割し、分割されたオブジェクトの地図上の絶対位置を、各空間領域の代表点からの相対位置に変換する。

各空間領域に沿って分割する処理は、例えば、オブジェクト A がオブジェクト詳細テーブル 2 3 では 1 レコード分存在するが、領域内オブジェクトテーブル 2 2 では 3 つの空間領域に分割されている。

【 0 0 7 9 】

S 5 0 4 において、オブジェクト登録部 4 0 は、作成した空間データ (領域索引ツリー 2 1、領域内オブジェクトテーブル 2 2) への登録データを、各空間データへと登録する

50

(詳細は、図12)。

【0080】

図12(a)は、オブジェクト登録部40が実行する、空間データ(領域索引ツリー21、領域内オブジェクトテーブル22)への登録処理(S504)の詳細を示すフローチャートである。

【0081】

S601において、オブジェクト登録部40は、S503で作成された登録対象のオブジェクトの外接矩形を検索範囲としたオブジェクト検索部30の処理(図7参照)を呼び出すことにより、登録対象の外接矩形のオブジェクトと交差する空間領域(交差領域)を求め、交差領域リストに挿入する。

10

S602において、オブジェクト登録部40は、交差領域リストの要素をすべて取得したか否かを判定する。判定条件を満たすときには本処理を終了し、判定条件を満たさないときにはS603へ進む。

【0082】

S603において、オブジェクト登録部40は、交差領域リストの先頭から順に、空間領域を1件取り出す。以下、取り出した空間領域を、選択領域と呼ぶ。

S604において、オブジェクト登録部40は、領域索引ツリー21の選択領域に対応するリーフノードからリンクする、登録済みのオブジェクトノードを取得する。

S605において、登録済みのオブジェクトの数と、今回の登録対象のオブジェクトの数との合計が、各空間領域のオブジェクトの容量(例えば、図2の場合、最大3つ)を超過するか否かを判定する。判定条件を満たすときにはS606へ、判定条件を満たさないときにはS607へ、それぞれ進む。

20

【0083】

S606において、領域分割部42は、1つの選択領域を4つの空間領域へと分割する。具体的には、まず、分割対象の選択領域を等面積に4分割した場合の4つの空間領域について、それぞれの代表点のモートン符号値を算出する。そして、領域索引ツリー21のリーフノードに登録されている選択領域の空間領域情報を削除するとともに、そのリーフノードにS611で算出した4つの空間領域の空間領域情報を追加する。

ここで、領域索引ツリー21のリーフノードの許容量(図2の場合、空間領域情報は最大3つまで)を超過したときには、図5に示すように、その超過したリーフノードを複数のリーフノードに分割することで、リーフノードの許容量を遵守する。このS606の処理により、空間領域の数が増えたことで、各空間領域のオブジェクトの容量に余裕が生まれ、S605の判定条件を満たしやすくなる。そして、処理をS503に戻す。

30

【0084】

S607において、相対座標値変換部41は、登録対象のオブジェクトノードが示すオブジェクトの外接矩形の位置を、絶対座標値から相対座標値に変換する(図12(b)参照)。

S608において、オブジェクト登録部40は、登録対象のオブジェクトを選択領域に登録する。具体的には、領域索引ツリー21の選択領域のリーフノードからリンクされるオブジェクトノードに登録対象のオブジェクトを追加するとともに、領域内オブジェクトテーブル22の選択領域を示すレコードに、登録対象のオブジェクトを追加する。

40

【0085】

図12(b)は、相対座標値変換部41が実行する、座標変換処理(S607)の詳細を示すフローチャートである。

【0086】

S611において、オブジェクトの外接矩形の絶対座標値である、左下座標値を(x_{min} , y_{min})、右下座標値を(x_{max} , y_{max})に設定する。

S612において、S611のオブジェクトが含まれる空間領域の代表点のモートン符号値および分割数を領域内オブジェクトテーブル22から読み取り、その空間領域の左下座標値(x'_{min} , y'_{min})と右上座標値(x'_{max} , y'_{max})とを算出する。

50

【 0 0 8 7 】

S 6 1 3 において、S 6 1 1 の左下座標値 (x_{\min} , y_{\min}) から S 6 1 2 の左下座標値 (x'_{\min} , y'_{\min}) を減算することで、オブジェクトの左下座標値の相対座標値 (x'_{\min} , y'_{\min}) を計算する。例えば、「 $x'_{\min} = x_{\min} - x'_{\min}$ 」である。

【 0 0 8 8 】

S 6 1 4 において、S 6 1 1 の右上座標値 (x_{\max} , y_{\max}) から S 6 1 2 の左下座標値 (x'_{\min} , y'_{\min}) を減算することで、オブジェクトの右上座標値の相対座標値 (x'_{\max} , y'_{\max}) を計算する。例えば、「 $x'_{\max} = x_{\max} - x'_{\min}$ 」である。

【 0 0 8 9 】

図 1 3 (a) は、オブジェクト削除部 5 0 が実行する、オブジェクトの削除のフローチャートを示す。

10

【 0 0 9 0 】

S 7 0 1 において、オブジェクト削除部 5 0 は、削除対象のオブジェクトの情報を取得する。S 7 0 1 の入力データは、登録対象のオブジェクトの ID が少なくとも入力されればよい。

【 0 0 9 1 】

S 7 0 2 において、オブジェクト削除部 5 0 は、S 7 0 1 で入力された削除対象のオブジェクトを、オブジェクト詳細テーブル 2 3 から削除する。具体的には、削除対象のオブジェクトの ID を検索キーとして、オブジェクト詳細テーブル 2 3 を検索し、該当するレコードをオブジェクト詳細テーブル 2 3 (表 3 参照) から削除する。

20

【 0 0 9 2 】

S 7 0 3 において、オブジェクト削除部 5 0 は、空間データ (領域索引ツリー 2 1、領域内オブジェクトテーブル 2 2) から、図 6 に示すように、削除対象のオブジェクトを削除する (図 1 3 (b) 参照) 。

【 0 0 9 3 】

図 1 3 (b) は、オブジェクト削除部 5 0 が実行する、空間データ (領域索引ツリー 2 1、領域内オブジェクトテーブル 2 2) からの削除処理 (S 7 0 3) の詳細を示すフローチャートである。

【 0 0 9 4 】

S 7 1 1 において、削除対象のオブジェクトの外接矩形に交差する空間領域の情報を交差領域リストに挿入する。なお、挿入する空間領域の情報は、領域内オブジェクトテーブル 2 2 (表 1 など参照) から削除対象のオブジェクトの ID をキーとして検索し、該当するレコードとして取得した情報である。

30

S 7 1 2 において、交差領域リストの要素 (空間領域) をすべて取得したか否かを判定する。判定条件を満たすときには本フローチャートを終了し、判定条件を満たさないときには S 7 1 3 へ進む。

S 7 1 3 において、交差領域リストから要素を 1 件取得し、選択領域とする。この選択領域に交差するオブジェクトは、領域内オブジェクトテーブル 2 2 の選択領域に対応するオブジェクトの列から取得することができる。

【 0 0 9 5 】

40

S 7 1 4 において、領域内オブジェクトテーブル 2 2 内の選択領域に対応する削除対象のオブジェクトを、領域内オブジェクトテーブル 2 2 から削除する。さらに、領域索引ツリー 2 1 の選択領域のリーフノードからリンクされるオブジェクトノードから、削除対象のオブジェクトを削除する。この削除処理により、選択領域内のオブジェクトの容量に 1 つ分余裕ができる。

【 0 0 9 6 】

S 7 1 5 において、選択領域を含む分割数が等しい、隣接する 4 つ組の空間領域 (例えば、図 6 ではモートン符号値 = 4 8、5 2、5 6、6 0 の 4 つ組の空間領域) に対応づけられているオブジェクト数の総和が容量数 (例えば、3 つ) 以下に収まるか否かを判定する。判定条件を満たすときには S 7 1 6 へ、判定条件を満たさないときには S 7 1 2 へ、

50

それぞれ進む。

【 0 0 9 7 】

S 7 1 6 において、領域併合部 5 1 は、S 7 1 5 で判定した 4 つ組の空間領域を 1 つの空間領域へと併合する。具体的には、以下の処理を行う。

- ・ 4 つ組の各空間領域（例えば、図 6 ではモートン符号値 = 4 8、5 2、5 6、6 0 の 4 つ組の空間領域）からリンクされるオブジェクトノードを、1 つのオブジェクトノードに併合するとともに、その併合前の各オブジェクトノードに含まれる各オブジェクトを（図 6 では、オブジェクト D、F、G）、併合したオブジェクトノードにまとめる。

- ・ 4 つ組の空間領域のうち、左下以外に存在する 3 つの空間領域情報（図 6 ではモートン符号値 = 5 2、5 6、6 0 の空間領域）を、領域索引ツリー 2 1 のリーフノードから削除する。その結果、空間領域情報をもたない（空の）リーフノードができたときには、そのリーフノードを削除するように、ツリー構造を変更する。

- ・ 4 つ組の空間領域のうち、左下に存在する空間領域情報（図 6 ではモートン符号値 = 4 8 の空間領域）を更新する。具体的には、空間領域情報からリンクされるオブジェクトノードを併合したオブジェクトノードとし、空間領域情報における分割数から 1 を減算する（図 6 では、分割数（空間領域情報の括弧内の数字）を「3」から「2」に減算している）。

【 0 0 9 8 】

以上説明した本実施形態によれば、オブジェクト登録部 4 0 がオブジェクトの形状および位置を多次元空間上に配置し、オブジェクト削除部 5 0 が配置されたオブジェクトを削除し、オブジェクト検索部 3 0 が入力された検索条件に該当するオブジェクトを検索する。そして、空間データ管理装置 1 は、オブジェクトが配置される空間データを、空間領域として領域索引ツリー 2 1 というバランス木のデータ構造で管理する。領域索引ツリー 2 1 のリーフノードでは、1 つの空間領域に対し 1 件の空間領域の情報が格納され、オブジェクトノードでは、オブジェクトの外接矩形がそのオブジェクトに重なる空間領域の代表点からの相対座標で表現されている。

これにより、領域索引ツリーの 2 1 の格納に必要なディスク容量を削減すると共に、オブジェクトの検索、登録、削除の各処理を高速で実現できる。よって、領域索引ツリー 2 1 の効果に伴う高速性かつ小容量性は、ハードウェア資源の使用制約が厳しい組込ソフトウェアにおいて、特に効果を発揮する。

【 符号の説明 】

【 0 0 9 9 】

- 1 空間データ管理装置
- 2 1 領域索引ツリー
- 2 2 領域内オブジェクトテーブル
- 2 3 オブジェクト詳細テーブル
- 3 0 オブジェクト検索部
- 3 1 検索初期設定部
- 3 2 交差判定部
- 3 3 検索起点設定部
- 3 4 絶対座標値変換部
- 4 0 オブジェクト登録部
- 4 1 相対座標値変換部
- 4 2 領域分割部
- 5 0 オブジェクト削除部
- 5 1 領域併合部
- 6 0 データ切替部
- 9 1 中央処理装置
- 9 2 主記憶装置
- 9 3 二次記憶装置

10

20

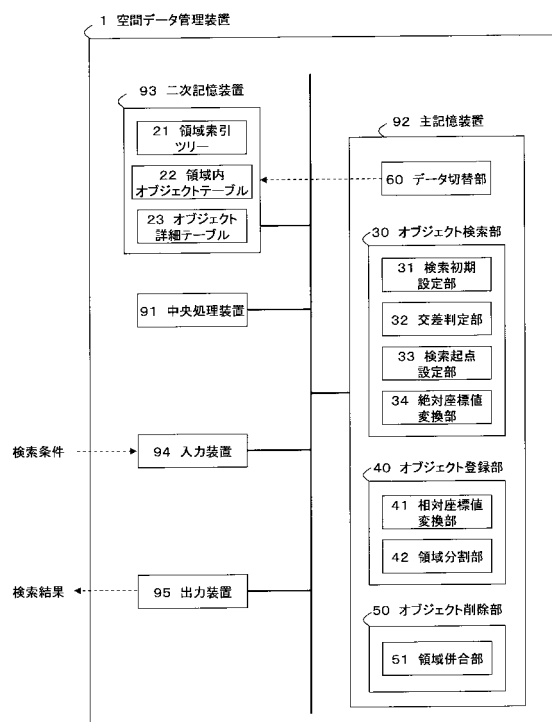
30

40

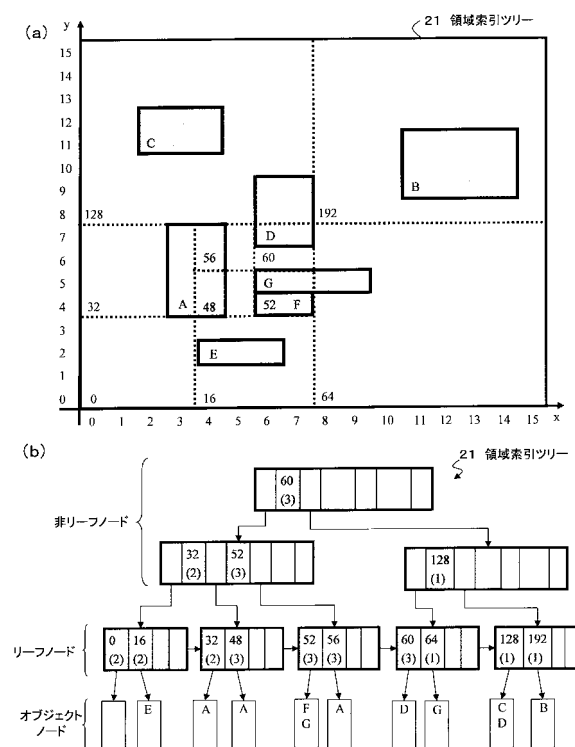
50

- 9 4 入力装置
 9 5 出力装置
 A , B , C , D , E , F , G , H , I オブジェクト

【図 1】

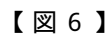
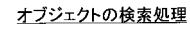


【図 2】

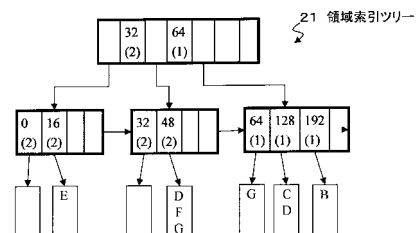
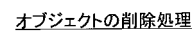


【 図 4 】

(a)



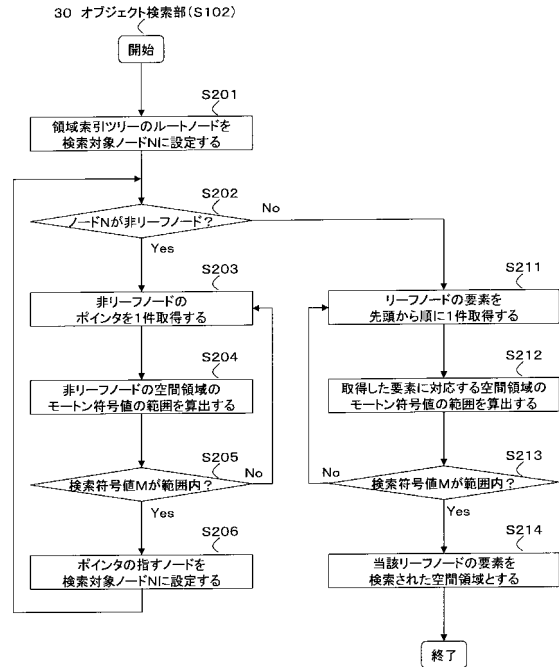
(a)



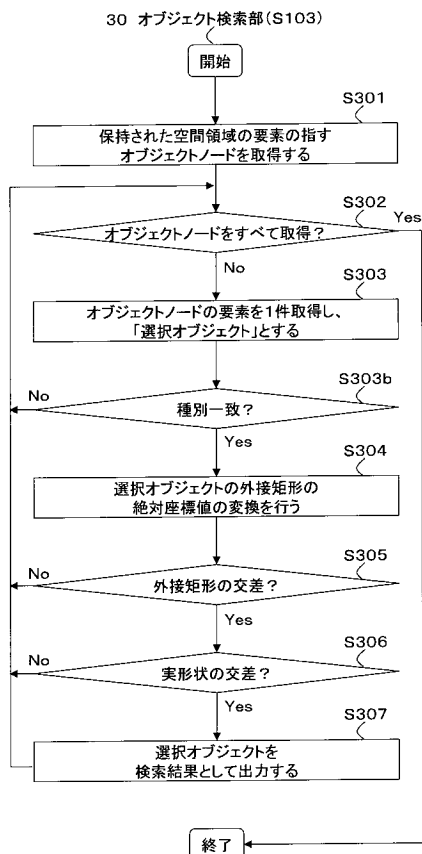
【図 7】



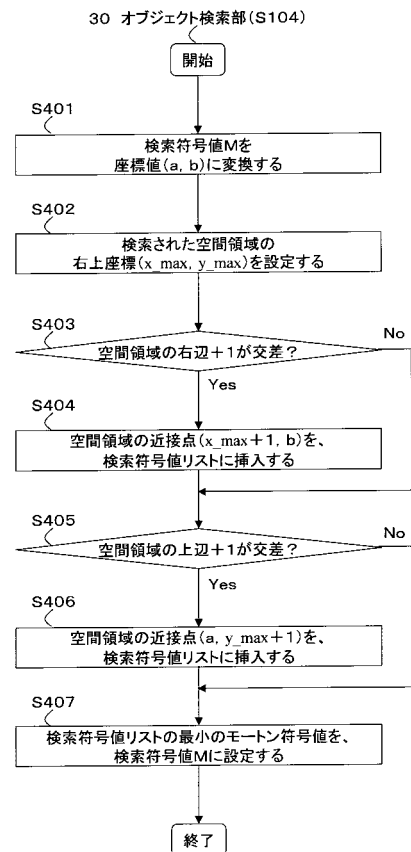
【図 8】



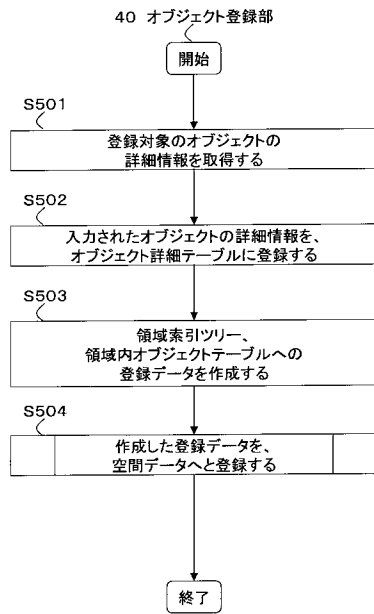
【図 9】



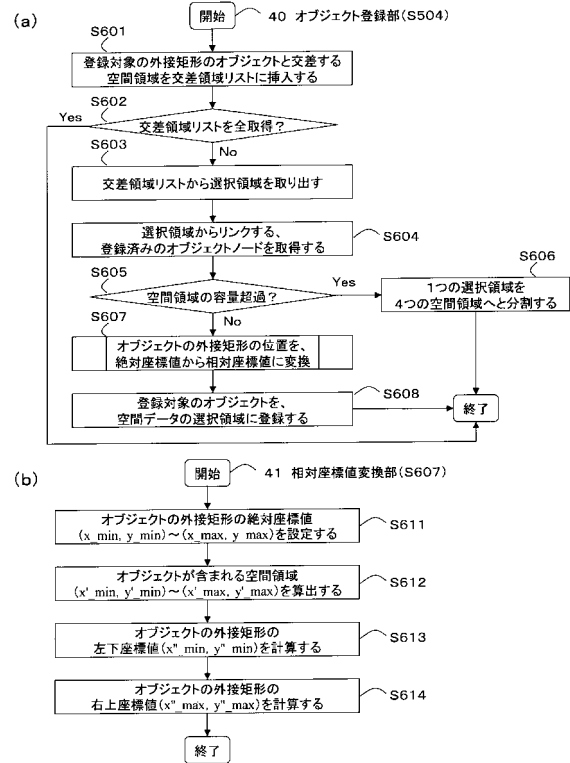
【図 10】



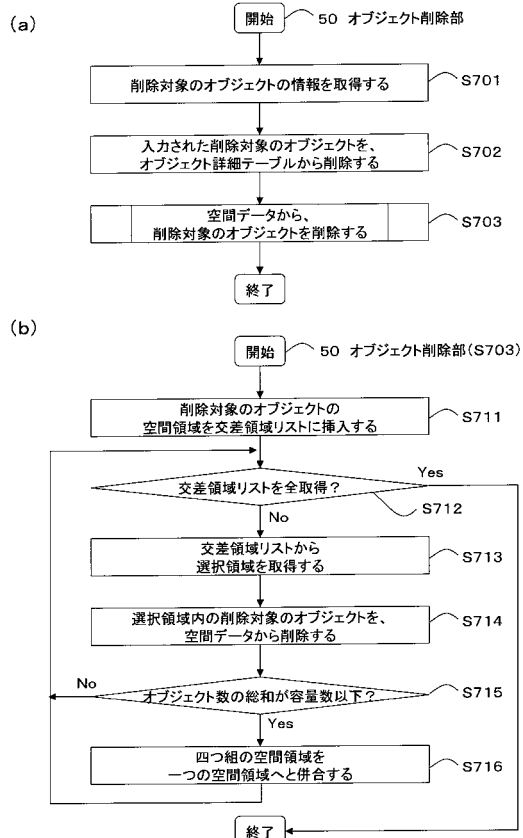
【図 1 1】



【図 1 2】



【図 1 3】



フロントページの続き

- (72)発明者 谷崎 正明
東京都国分寺市東恋ヶ窪一丁目280番地 株式会社日立製作所 中央研究所内
- (72)発明者 木村 耕治
神奈川県横浜市戸塚区戸塚町5030番地 株式会社日立製作所 ソフトウェア事業部内
- (72)発明者 梶山 尚紀
東京都品川区東品川4丁目12番7号 日立ソフトウェアエンジニアリング株式会社内
- (72)発明者 入江 正憲
東京都品川区東品川4丁目12番7号 日立ソフトウェアエンジニアリング株式会社内

審査官 鈴木 和樹

- (56)参考文献 特開平5-242164(JP, A)
特開2005-135440(JP, A)
特開平11-328225(JP, A)
ASHRAF ABOULNAGA、外1名, Window Query Processing in Linear Quadrees, Distributed and Parallel Databases [online], 2001年 9月 1日, 第10巻, 第2号, p. 111-126, URL, http://www.cerias.purdue.edu/apps/reports_and_papers/view/3829
乗松誠志、外2名, Quad Treesに対する比較実験と考察, 情報処理学会研究報告(90-AL-17), 日本, 社団法人情報処理学会, 1990年 9月29日, 第90巻, 第79号, p. 35-44
柳沼良知、外1名, 外接長方形上の点を用いた図形の間接表現の一提案, 電子情報通信学会論文誌, 日本, 社団法人電子情報通信学会, 1992年 9月25日, 第J75-D-I巻, 第9号, p. 821-829
CLIFFORD A. SHAFFER、外2名, QUILT: a geographic information system based on quadrees, INT.J.GEOGRAPHICAL INFORMATION SYSTEMS [online], 1990年, 第4巻, 第2号, p. 103-131, URL, <http://people.cs.vt.edu/shaffer/Papers/IJGIS90.pdf>

- (58)調査した分野(Int.Cl., DB名)
G06F 17/30