

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第4259864号
(P4259864)

(45) 発行日 平成21年4月30日(2009.4.30)

(24) 登録日 平成21年2月20日(2009.2.20)

(51) Int.Cl. F I
G O 6 F 17/30 (2006.01) G O 6 F 17/30 4 1 9 B

請求項の数 7 (全 29 頁)

(21) 出願番号	特願2002-550552 (P2002-550552)	(73) 特許権者	390028587
(86) (22) 出願日	平成13年12月11日(2001.12.11)		ブリティッシュ・テレコミュニケーションズ・パブリック・リミテッド・カンパニー
(65) 公表番号	特表2004-516552 (P2004-516552A)		BRITISH TELECOMMUNICATIONS PUBLIC LIMITED COMPANY
(43) 公表日	平成16年6月3日(2004.6.3)		イギリス国, イーシー1エー・7エー ジェイ, ロンドン, ニューゲート・ストリート 81
(86) 国際出願番号	PCT/GB2001/005479		
(87) 国際公開番号	W02002/048908	(74) 代理人	100058479
(87) 国際公開日	平成14年6月20日(2002.6.20)		弁理士 鈴江 武彦
審査請求日	平成16年11月2日(2004.11.2)	(74) 代理人	100091351
(31) 優先権主張番号	00311228.1		弁理士 河野 哲
(32) 優先日	平成12年12月15日(2000.12.15)	(74) 代理人	100088683
(33) 優先権主張国	欧州特許庁 (EP)		弁理士 中村 誠
(31) 優先権主張番号	00311234.9		
(32) 優先日	平成12年12月15日(2000.12.15)		
(33) 優先権主張国	欧州特許庁 (EP)		

最終頁に続く

(54) 【発明の名称】 エンティティを検索する方法

(57) 【特許請求の範囲】

【請求項 1】

所定の領域内に包含されている 1 又はそれ以上の点を検索する方法であって、該方法はプログラムされたプロセッサによって実行され、

(i) 1 又はそれ以上の点を取囲んでいる区域を識別するデータをデータ記憶装置から検索するステップであって、該区域はリンクデータと関係していて、リンクデータは各区域について、該区域により取囲まれた点を識別するものである、ステップと；

(ii) 該区域について、

該区域の広がり程度の程度を該所定の領域の広がり程度の程度と比較して、該区域が該所定の領域と重なるかどうかを確立するステップと、

重なりがある場合には、該区域の部分区域を識別するデータを該データ記憶装置から検索し、この部分区域のいずれかであってその広がり程度の程度が該所定の領域の中に完全に入っているものを識別するステップとを含むプロセスを実行する、ステップと；

(iii) 各部分区域について、該所定の領域内部に完全に入るすべての部分区域が識別されるまで上記プロセスを繰返すステップと；

(iv) 該識別された部分区域に対応しているリンクデータにアクセスして、該部分区域により取囲まれた点を検索するステップとを備え、

該複数の点は部分区域間の関係に従って点のリストとして予め記憶されており、また該リンクデータは該点のリスト内の対応している取囲まれた点の第一のものの位置を示している値を含んでおり、また、

該ステップ (iv) は該識別された部分区域の各々について、

その部分区域によって取囲まれた点の数を表わしている識別子を検索し、該識別された部分区域と関係している位置の値を検索するステップと、

該点のリストにアクセスして該位置の値によって与えられた該リスト内の位置から該その部分区域によって取囲まれた点の数を検索するステップとを備えている、方法。

【請求項 2】

請求項 1 記載の方法であって、該区域と部分区域とを識別するデータと、対応しているリンクデータとがインデックスで記憶される方法。

【請求項 3】

請求項 1 または 2 記載の方法であって、該所定の領域は、レンジ情報に対応している方法。 10

【請求項 4】

請求項 3 記載の方法であって、該レンジ情報は地理的なレンジ情報、動作時間情報、配信時間情報及び医用情報のいずれか 1 つを含んでいる方法。

【請求項 5】

計算機に請求項 1 乃至 4 のいずれか 1 項に記載の方法のステップを実行させるためのプログラム。

【請求項 6】

検索要求を送信するステップを備えた検索方法であって、該検索要求は点が検索されることになるある領域を示している信号とトリガ信号とを含んでおり、該トリガ信号は請求項 1 乃至 4 のいずれか 1 項記載の方法を発動させる検索方法。 20

【請求項 7】

所定の領域により包含されている 1 又はそれ以上の点を検索する装置であって：該装置は、

該所定の領域を示す信号を含んでいる検索要求を受信するように構成された入力手段と；

1 又はそれ以上の区域を示すデータであって、各区域は 1 又はそれ以上の点を取囲んでいるデータと；該区域と関係しているリンクデータであって、リンクデータの各アイテムはそれぞれの区域により取囲まれた点を識別するリンクデータと；該点を識別するデータとを記憶するように構成された記憶手段と； 30

該記憶手段にアクセスして該所定の領域に対応している 1 又はそれ以上の点を検索するように構成された検索手段と；

該 1 又はそれ以上の検索された点を出力するように構成された出力手段とを備えており、

該装置は、検索要求の受信にตอบสนองして、該検索手段が該記憶手段からある区域を検索して、該検索した区域について、

該区域の広がり の程度を該所定の領域の広がり の程度と比較して、該区域が該所定の領域と重なるかどうかを確定する段階と；

重なりがある場合には、該領域の部分区域を識別するデータを検索し、該所定の領域内部に広がり の程度が完全にあるいずれかの部分区域を識別する段階とを含むプロセスを実行し、 40

次に、該検索手段は、各部分区域について、完全に該所定の領域内に入るすべての部分区域が識別されるまで、該プロセスを繰返して、該識別した部分区域に対応しているリンクデータにアクセスし、該部分区域により取囲まれた点を検索して、それにより該出力手段が該検索した点を出力する、ように構成されており、

前記記憶手段内の該複数の点は部分区域間の関係に従って点のリストとして予め記憶されており、また該リンクデータは該点のリスト内の対応している取囲まれた点の第一のものの位置を示している値を含んでおり、また、

該検索手段は該識別した部分区域の各々について、

その部分区域によって取囲まれた点の数を表わしている識別子を検索し、該識別した 50

部分区域と関係している位置の値を検索し、

該点のリストにアクセスして該位置の値によって与えられた該リスト内の位置から該部分区域によって取囲まれた点の数を検索するようにさらに構成されている、装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

この発明は、検索 (retrieval) 方法に係り、データベースもしくは同等の記憶装置 (メモリ) 内に記憶されたエンティティ (entities) とともに使用するのに適している。

【0002】

【従来の技術】

あるデータベース内に莫大な数のエンティティが存在しているときには、そのデータベース内のデータに関する問合せと合理的な時間的期間の中で調和をとりながらエンティティを識別することが決して些細な活動でないことはすぐに分ることである。検索プロセスを容易にするために、データベース内のデータは、一般に何らかの方法でインデックスを付けられ、問合せはそこでインデックスについて実行される。エンティティがインデックスを付けられるやり方は検索することの品質と速度について重要な意味をもつと期待できるのであり、情報がデータベース内に次第に増すように記憶されると、データにインデックスを付ける改良方法を見付けることが重要な関心事となる。

【0003】

場所の名称に基づいて、位置データにインデックスを付けることは知られている。また場所の名称から地理的な一組の座標を検索して、この座標から抽出された地形学的 (topological) 情報に基づいたインデックスを構築することも知られている (例えば、“GIPSY” は UC バークレイ校でジョイント NSF / NASA / ARPA (Wilensky 他, 1994) 主導で共同開発されている)。さらに、地理的な座標自体に基づいてインデックスを構築することも知られている。Oracle™ のようなデータベースベンダ (vendors) は幾何学的なデータを記憶してインデックスを付けるためのシステムを開発しており、例えば Oracle 空間データカートリッジがそれであり、これは空間的な問合せが SQL の拡張された (標準形ではない) 形式を用いて実行されるようにしている。その他のベンダは MapInfo™, SpatialWare™, Innogistic™, 及び Informix™ のようなもので、空間データを処理するのに類似の専有のやり方を有している。とくに、Innogistic™ は Cartology DSI として知られている製品を開発し、これが幾何学的なベクトルデータをプロブ (blobs) として記憶する (これは二値の大きなオブジェクトであり、基礎に置かれたデータベースによっては真性な (intrinsically) ものとして認識できないものである。Innogistic™ はよく知られた “クアッド - ツリー (quad tree)” というアイデアに基づいてデータベースの外部でインデックスを創ることもする。このインデックスデータはバイナリ - ツリー (binary - tree) の構造の中に記憶されて、Distributed Component Object Model (DCOM) ミドルウェアサービスによってアクセスされる。

【0004】

【発明が解決しようとする課題】

Oracle™ と Innogistic™ の両システムは、クアッド - ツリー方法の使用を行ない、この方法ではレイヤ (layer) の全領域が分割されまた部分分割 (subdivided) されて一連の四つがネストした (nested) 正方形とされる。全体の領域は 0, 1, 2, 3 と指定された四つの正方形の一つに対して指定されている。こういった正方形の各々は四つのもっと小さな正方形に部分分割される。正方形 1 の領域は 10, 11, 12, 13 となる。こういったものの各々がさらに部分分割され、これが意味しているのは、例えば正方形 11 の部分分割がインデックス値 110, 111, 112, 113 を指定されるということである。結果として、地図の中のいずれの場所も単一のインデックス番号によって参照することができる。このクアッドツリーの方法での不都合な点は、部分分割された正方形の内部に点が一つも存在しないとすると処理時間が浪費されるということである。もしインデックス付けが大きな領域にわたって実行されるとすると、この浪費した処理時間は些細なものと

10

20

30

40

50

はならずコスト高のものとなる。

【0005】

【課題を解決するための手段】

この発明の第一の特徴によると、所定の領域により包含されている点 (points) を検索する方法が提供されている。この方法は、

(i) 1又はそれ以上の点を取囲む区域を識別するデータを検索する段階であって、該区域はリンクデータと関係していて、リンクデータは各区域について、該区域により取囲まれた点を識別する段階と；

(ii) 該区分についてプロセスを実行する段階と；

該プロセスは、該区域の広がり の程度を該所定の領域の広がり の程度と比較して、該区域が該所定の領域と重なっているかどうかを確立する段階と、重なりがある場合には、該区域についての部分区域を識別し、この部分区域のいずれかであってその広がり の程度が該所定の領域の中に完全に入っているものを識別する段階とを有し、

(iii) 各部分区域について、該所定の領域内部に完全に入るすべての部分区域が識別されるまで上記プロセスを繰り返す段階と；

(iv) 該識別された部分区域に対応しているリンクデータにアクセスして、該部分区域により取囲まれた点を検索する段階とを備えている。

【0006】

該複数の点が該区域と部分区域との間の所定の関係により与えられる順序で、点のリストとして有効に予め記憶されている。さらに、リンクデータは該点のリスト内の対応している取囲まれた点の第一のものの位置を示している値を含んでいる。

【0007】

また、該アクセスする段階は、取囲まれた点の数を表わしている識別子を検索することと、該識別された区域または部分区域と関係している位置の値を検索することとを含んでいる。これが該位置の値と関係している該リスト中の位置から、取囲まれた点の数を検索できるようにする。

【0008】

便利なのは、点が二次元で表現できるデータと対応していることで、例えば位置 (緯度と経度) データまたはレンジデータであることである。レンジデータは、営業時間 (例えば始業及び終業時刻)、価格のマージン (例えば最高及び最低価格)、及び/又は医用データ (例えば、最高と最低血圧) を含んでいる。したがって、所定の領域は価格のレンジとすることができ、最高の家の価格と最低の家の価格といったものとすることができる。上で記述した方法によると、所定の領域の広がり は (すなわち、価格レンジ) はインデックスから検索された領域と比較される。特定の価格レンジの内に完全に入るある区域が識別されるまでこの価格のレンジと重なっているすべての領域はそこで継続して検索され、この識別された区域内に入るすべての点は、こうして特定された最大と最小の価格レンジ内に入るある価格をしている商品を表わす。

【0009】

【実施の形態】

この発明のさらなる様相、特徴、利点は添付の図面を参照して以下の好ましい実施形態の記述から明らかになる。

【0010】

概要

図1に示したようなデータベースサーバDB1, DB2は一般にユーザによる検索のための情報を記憶する。物理的なレベルでは、データベースサーバDB1, DB2が置かれている通信環境は少くとも一つのユーザインターフェースを含み、これが普通は計算機端末又はワークステーションT3により用意される。

【0011】

この発明の実施形態は、ワークステーションT3上で実施することができ、ワークステーションT3はデータベースサーバDB1, DB2に接続されている。データベースサーバ

10

20

30

40

50

DB1, DB2は端末T3と同じLAN N1上に示されてはいるが、データベースサーバDB1, DB2は異なる網に接続されるようにできて、異なる網が今度は1又はそれ以上のスイッチ及び/又はルータ(図示せず)を經由してLAN N1に接続されるようになると理解される。実施形態は入力として、例えばファイルとしてデータを受領して、データに対するインデックスを構築するが、その詳細は後述するところである。構築されたインデックスはそこでデータベースDB1, DB2の一つの中にセーブ(保管)され、またインデックスを付けたデータもセーブされ、その順序は構築されたインデックスの構造により与えられ、データベースDB1, DB2の一つにセーブされる。構築されたインデックスは、そのデータが記憶されていたデータベースと同じか異なるデータベース上にセーブすることができる。

10

【0012】

この発明の実施形態は、二次元で定義されているエンティティにインデックス付けをすることに関係している。実施形態によりインデックス付けができるエンティティの明白な例にはオブジェクト(objects)の場所、例えば給油所、現金出納機等のあるところであって、オブジェクトの位置は通常は緯度と経度とによって定義されているところである。数多くの他のエンティティが二次元によって提示できる。例えば、モーターバイクの加速度は時間と速度との関数であり、ある材料の伝導率は材料の性質と温度との関数であり、オブジェクトの変形は材料の性質とオブジェクトに加えられる力との関数である等という具合である。さらに、変換をn次元パラメータに適用してこのパラメータを二次元パラメータにすることができて、二次元パラメータを二次元空間で表示することができる。

20

【0013】

この発明の実施形態を用いてインデックス付けができるエンティティのさらなる例はレンジ情報を含んでいて、例えば時間的な情報、価格情報及び医用状態情報などを含んでいる。

【0014】

時間的なレンジ(range)情報の例は営業(business)とかレジャー施設の開始と終了の時間であり、こういった時間は二次元で表現できる。例えば終了と開始の時刻とがそれぞれ縦軸と横軸とで表現できる。同じように、一番速い時刻と一番遅いデリバリー時刻とが二次元で表現できる。

【0015】

価格情報の例は商品の価格を含み、したがって、例えば商品の最高価格と最低価格とが二次元の一方上でそれぞれ表現できて、この発明の実施形態を用いてインデックス付けができる。価格情報はまた取引情報を含んでいて、株式、債券などの売買に使用されることとなる。

30

【0016】

医用状態情報の例は、体温と血圧といった測定可能な状態に関する統計を含んでいるし、癌の部位といった数値表示に変換可能な状態を含んでいる。

【0017】

以下の記述では、エンティティは一般に“点(points)”と呼ばれ、エンティティのコンテキスト(例えば給油所、現金出納機など)を実施形態の機構から分離するのにあてている。

40

【0018】

全体を概観するとして、点にインデックスを付ける方法の実施例は、図2を参照して記述される。図2は領域R1を示し、この領域の内部では、複数の点200が置かれている。各点はx, y座標空間で定義される。本質的に、領域R1がインデックスを付けられることになる点を含んでいて、領域R1が調べられて、点を含んでいる領域R3, R4と、点を含んでいない領域R2に分割される(領域はどんな形に分けることもでき、矩形、三角形、条(strips)のようになってよいが、図2はこの発明の発明思想を記述する目的で条に分けられた領域を示している)。この実施形態は、次に領域R3とR4との中の点の分布を調べ、領域R1について考慮されたよりも小さな規模(scale)についての識別し、

50

R 3 と R 4 との領域には点が含まれている。図 3 を参照すると、R 4 は本質的に R 1 1 となり、R 1 1 内部での点の分布が調べられる。このやり方で点の分布に集中してみると、いずれの点も含んでいない領域である、図 2 の領域 R 2 と、図 3 の領域 R 1 2 とは黙示的に無視されている。プロセスは継続して繰返されて、小さくなって行くサイズの一連の領域を介して実効的に“掘り下げ (burrowing down)”が進み、領域のサイズが単一の点のサイズを破壊してしまうところまで行く。実施形態の“掘り下げ”として、各領域がその上の領域とリンクされていて、それにより各点は一連の領域によってリンクされている。こういった点に対するインデックスは、一連の領域を含んでいて、こういった領域と点とはインデックスを創るために使用される (後により詳述する) が、このインデックスがデータベース DB 1 内にセーブされる。点と領域との間の関係はインデックス内の領域の識別子によって識別されることになる。

10

【 0 0 1 9 】

上述したように、インデックスを創る利点の一つは問合せ探索ドメインが領域に限られていて、この領域は点を含むことが既知であることである。すなわち、問合せはデータベース DB 1 内にセーブされた領域上でのみ実行されることになり、またこういった領域は定義により点を含んでいるので、探索ドメインは相対的にコンパクトなものとなっている。もう一度図 2 の参照に戻ると、ある問合せに関して点を識別するプロセスは上述の実施形態によると、領域 R 1 全体に係る情報を備えたインデックスの場合よりも高速となる。

【 0 0 2 0 】

実施形態の特定形式が図 4 ~ 1 2 を参照して提示されているが、そこでは、点は x, y 空間内の二次元座標となっている。インデックスを付けることになるエンティティが加速度値であるとすると、この加速度値は時間と速度との値についての対応している組によって定義されていて、この時間と速度との値は x, y 座標空間上に直接写像されて、それにより $(t_1, v_1), (t_2, v_2) \dots (t_n, v_n)$ が加速度値に対応している点の座標となる。同様に、もしインデックスを付けることになるエンティティが位置の値であって緯度と経度との値の対応する組により定義される値であれば、緯度と経度との値が x, y 座標空間上に直接写像されて、 $(lat_1, long_1) \dots (lat_n, long_n)$ が位置の値に対応している点の座標となる。点は記憶されている (例えばあるファイルに書込まれている) と仮定されていて、それによりこの発明の実施例はファイルから点を読み出す。代りの実施例ではユーザが点を入力できるものとし、その条件として点についてのインデックスが構築されそうになっているときとしている。

20

30

【 0 0 2 1 】

さらに、以下に示す実施例では、領域は正方形であって、以下の記述では“クアッド (quads)”とが“サブクアッド (sub-quads)”と呼ぶこととし、各クアッド (quad) は継続して四つのサブクアッド (sub-quad) に分割されている。各サブクアッドは点の存在について調べられる。点のないものは無視されて、このことは図 2 と図 3 とを参照して記載された領域 R 2 を無視することと同じプロセスであり、各新しいサブクアッドで点を含んでいるものがそのサブクアッド内の点を含んでいる一番小さな領域の周囲に“シュリンク・ラップ (shrunkwrapped)”される。この実施例は正方形に従って領域を解析するが、数多くの他の形状も点の周りで“シュリンク・ラップ (shrink-wrap)”するのに使用できる。各サブクアッドは次に再び四つのサブクアッドに分けられて、空のサブクアッドが再び無視されて、各残っているサブクアッドが点を含んでいる一番小さな領域の周りで“シュリンク・ラップ”される。結果的に、各残っているサブクアッドは単一の点に“シュリンク・ラップ”されて、その座標が関与している点の座標となる。一旦クアッドとすべてのサブクアッドであって、無視されずに介在しているサブクアッドと単一の点と一致しているサブクアッドの両方を含んでいるものが識別されると、その点に対するインデックスでそれぞれのものと関連のあるクアッドとサブクアッドを備えているものが作られる。これが図 6 ~ 図 1 1 を参照して後に詳しく論じられる。

40

【 0 0 2 2 】

以下では、段階 S 4 . 1 ないし S . 4 . 9 が図 4 と図 5 との流れ図で連続して示されてお

50

り、また図 6 ないし 12 で示すように同じ参照番号で動作により示されている。

【 0 0 2 3 】

図 6

・段階 S 4 . 1 インデックスを付けることになるすべての点の x , y 座標を読み込む。
(上述したように、これらの点は記憶場所であるファイルなどから読出されてよいし、あるいはユーザから直接であってもよい) ;

・段階 S 4 . 2 すべての点について境界を形成している (bounding) 箱を製図し、識別し、その周りを “ シュリンクラッピング (shrink wrapping) ” する。 x 次元、 y 次元の両方における一番外側の点の座標間の差、それぞれ $d x$ と $d y$ によって境界を形成している箱が与えられる。この箱は一番外側のクアッド 5 0 1 である ;

・段階 S 4 . 3 及び S 4 . 4 クアッド 5 0 1 の広がり (extent) をセーブする。すなわち $d x$, $d y$ をセーブし、その中の点の座標をセーブする ;

・段階 S 4 . 5 一番外側のクアッド 5 0 1 は正の寸法を有しているか (すなわち、クアッド 5 0 1 の $d x$ と $d y$ とはゼロに等しいか ?) どうかをチェックする。図 6 に示した例では、一番外側のクアッド 5 0 1 が存在し、その理由はこのクアッドがその中に複数の点を有していて、 $d x$, $d y$ はゼロでなく、したがってクアッド 5 0 1 が正の寸法 (size) をもつことによる ;

・段階 S 4 . 6 一番外側のクアッド 5 0 1 を四つのサブクアッド 5 0 3 a , 5 0 3 b , 5 0 3 c , 5 0 3 d に分割する ;

・段階 S 4 . 6 . 1 それぞれのサブクアッドで各点をタグ付けして、各サブクアッド内の点の数を記録する ;

・段階 S 4 . 7 サブクアッド 0 (5 0 3 a) から始まって、サブクアッド 5 0 3 a 内にいずれかの点が存在するかどうかをチェックする。
点があるときには、このサブクアッド 0 (5 0 3 a) 内部の点を段階 S 4 . 1 に対して入れて、段階 S 4 . 1 を介してサブクアッド 0 (5 0 3 a) について進行させることとし、図 7 を参照して後述するように、実行する。

【 0 0 2 4 】

図 7

・段階 S 4 . 1 サブクアッド 5 0 3 a に対応している点の x , y 座標を読み込む ;

・段階 S 4 . 2 5 0 3 a 内のすべての点について境界を形成している箱を製図し、 “ シュリンクラップ ” したサブクアッド 0 (6 0 1) を創る。これが上述した原理を示していて、実施例はサブクアッド 0 内部の領域で点のないところを識別し、この領域はそこで無視される ;

・段階 S 4 . 3 及び段階 S 4 . 4 サブクアッド 0 (6 0 1) の広がり (extent) の程度すなわち、 $d x$, $d y$ と、その中の点の座標とをセーブする ;

・段階 S 4 . 5 サブクアッド 6 0 1 が正の寸法をもつかどうか (すなわち、クアッド 6 0 1 の $d x$, $d y$ がゼロに等しいかどうか ?) をチェックする。図 7 で見るように、クアッド 6 0 1 はその中に複数の点を有していて、クアッド 6 0 1 については $d x$, $d y$ はゼロではなく、したがってサブクアッド 6 0 1 は正の寸法 (サイズ) を有している ;

・段階 S 4 . 6 サブクアッド 0 (6 0 1) を四つのサブクアッド、すなわち 0 , 0 (6 0 3 a) , 0 , 1 (6 0 3 b) , 0 , 2 (6 0 3 c) , 0 , 3 (6 0 3 d) に分割する ;

・段階 S 4 . 6 . 1 各点に関連するサブクアッドでタグ付けをして、各サブクアッド内の点の数を記録する ;

・段階 S 4 . 7 サブクアッド 0 , 0 (6 0 3 a) から始まって、サブクアッド 0 , 0 (6 0 3 a) 内にいずれかの点が存在するかどうかをチェックする : 点があるときには、このサブクアッド 0 , 0 (6 0 3 a) 内部の点を段階 S 4 . 1 に入れて、段階 S 4 . 1 を介してサブクアッド 0 , 0 (6 0 3 a) について進行させることとし、このことは図 8 を参照して後述する。

【 0 0 2 5 】

図 8

10

20

30

40

50

- ・段階 S 4 . 1 サブクアッド 6 0 3 a に対応している点の x , y 座標を読み込む ;
- ・段階 S 4 . 2 サブクアッド 0 , 0 (6 0 3 a) 内のすべての点について境界を形成している (取囲んでいる) 箱を製図し、 “ シュリンクラップ ” したサブクアッド 0 , 0 (7 0 1) を創る。サブクアッド 0 について、サブクアッド 6 0 3 a 内部で点がない領域は無視される ;
- ・段階 S 4 . 3 及び段階 S 4 . 4 サブクアッド 0 , 0 (7 0 1) の広がり の程度すなわち、 $d x$, $d y$ をセーブして、その中の点の座標をセーブする ;
- ・段階 S 4 . 5 サブクアッド 7 0 1 が正の寸法を持っているかどうか ? をチェックする (すなわち、クアッド 7 0 1 の $d x$, $d y$ がゼロに等しくないか ? をチェックする) 。 図 8 から分るように、クアッド 7 0 1 はその中に複数の点を有していて、クアッド 7 0 1 について $d x$, $d y$ はゼロではなく、このサブクアッド 7 0 1 は正の寸法をもつ。
- ・段階 S 4 . 6 サブクアッド 0 , 0 (7 0 1) を四つのサブクアッド : 0 , 0 , 0 (7 0 3 a) 、 0 , 0 , 1 (7 0 3 b) 、 0 , 0 , 2 (7 0 3 c) 、 0 , 0 , 3 (7 0 3 d) に分割する。
- ・段階 S 4 . 6 . 1 各点に関連のサブクアッドでタグ付けをして、各サブクアッド内の点の数を記録する ;
- ・段階 S 4 . 7 サブクアッド 0 , 0 , 0 (7 0 3 a) で開始して、サブクアッド 0 , 0 , 0 (7 0 3 a) 内にいずれかの点が存在するかどうかチェックする : サブクアッド 0 , 0 , 0 (7 0 3 a) 内に一つの点が存在すれば、このサブクアッド 0 , 0 , 0 内部の点を段階 S 4 . 1 に入れて、段階 S 4 . 1 を介してサブクアッド 0 , 0 , 0 (7 0 3 a) について進行することとし、それについては図 8 を参照してさらに以下で記述する。

【 0 0 2 6 】

図 8 についても

- ・段階 S 4 . 1 サブクアッド 7 0 3 a に対応している点を x , y 座標で読み込む ;
- ・段階 S 4 . 2 サブクアッド 0 , 0 , 0 (7 0 3 a) 内のすべての点について取囲んでいる箱を製図して、 “ シュリンクラップ ” したサブクアッド 0 , 0 , 0 を作る : ここでは単一の点の周りに作ることになる。
- ・段階 S 4 . 3 及び段階 S 4 . 4 “ シュリンクラップ ” したサブクアッド 0 , 0 , 0 の広がり の程度をセーブし、ここでは単一点となるまでにしてあるので $d x$ と $d y$ とはゼロに等しく、この点の座標をセーブする ;
- ・段階 S 4 . 5 この点をもつサブクアッドが正の寸法を有するかどうかをチェックする。(すなわちこの点の $d x$ と $d y$ がゼロに等しいかどうかをチェックする。) 事実 $d x$ と $d y$ とは共にゼロであり、その理由はサブクアッド 7 0 3 a が単一の点にまで分解されていることによる。そこで段階 S 4 . 8 へ進む ;
- ・段階 S 4 . 8 サブクアッドカウンタ i をこのレベル (0 , 0 , i) でインCREMENT し、サブクアッド 0 , 0 , 1 (7 0 3 b) 内部の点を段階 S 4 . 1 に入れて (段階 S 4 . 7) 、段階 S 4 . 1 を介してサブクアッド 0 , 0 , 1 (7 0 3 b) について進行する。これについては図 9 を参照して以下に記述する。

【 0 0 2 7 】

図 9

- ・段階 S 4 . 1 サブクアッド 7 0 3 b に対応している点の x , y 座標を読み込む ;
- ・段階 S 4 . 2 サブクアッド 0 , 0 , 1 (7 0 3 b) 内のすべての点について取囲んでいる箱を製図して、 “ シュリンクラップ ” したサブクアッド 0 , 0 , 1 (8 0 1) を創る。サブクアッド 0 についてと同様に、サブクアッド 7 0 3 b 内部で点のない領域は無視される ;
- ・段階 S 4 . 3 及び段階 S 4 . 4 サブクアッド 8 0 1 についての広がり の程度すなわち、 $d x$, $d y$ をセーブし、またその中の点の座標をセーブする ;
- ・段階 S 4 . 5 サブクアッド 8 0 1 が正の寸法かどうか (すなわち、クアッド 7 0 3 a の $d x$, $d y$ がゼロに等しいかどうか) をチェックする。サブクアッド 8 0 1 がその中に複数の点をもっていれば、クアッド 8 0 1 についての $d x$, $d y$ はゼロでなく、したがっ

てサブクアッド801は正の寸法を有する；

・段階S4.6 サブクアッド0,0,1(801)を四つのサブクアッド、0,0,1,0(803a)、0,0,1,1(803b)、0,0,1,2(803c)、0,0,1,3(803d)に分割する。

・段階S4.6.1 関連するサブクアッドで各点をタグ付けし、各サブクアッド内の点の数を記録する；

・段階S4.7 サブクアッド0,0,1,0(803a)で開始して、サブクアッド0,0,1,0(803a)内にいずれかの点があるかどうかをチェックする：サブクアッド0,0,1,0(803a)には点は存在しない；

・段階S4.8 サブクアッドカウンタ*i*をこのレベル(0,0,1,*i*)でインCREMENTして、サブクアッド0,0,1,1(803b)内部の点を段階S4.1に加えて(段階S4.7)、段階S4.1を介して、サブクアッド0,0,1,1(803b)について進行していく。このことについては図10を参照して以下で記述する。

【0028】

図10

・段階S4.1 サブクアッド803bに対応している点の*x*,*y*座標を読み込む；

・段階S4.2 サブクアッド0,0,1,1(803b)内のすべての点について取囲んだ箱を製図し、“シュリンクラップ”したサブクアッド0,0,1,1を作り、これが単一の点となる；

・段階S4.3及び段階S4.4 この単一の点の広がり(extent)、すなわち*d*_{*x*},*d*_{*y*}をセーブし、この点の座標をセーブする；

・段階S4.5 この点が正の寸法をもっているかどうか、すなわちこの点の*d*_{*x*},*d*_{*y*}がゼロに等しいかどうかをチェックする。事實は、サブクアッド803bが単一の点に分解されているので、*d*_{*x*}も*d*_{*y*}も共にゼロであり、段階S4.8に進む；

・段階S4.8 このレベル(0,0,1,*i*)でサブクアッドカウンタ*i*をINCREMENTし、サブクアッド0,0,1,2(803b)内部の点を段階S4.1に入れて、段階S4.1を介してサブクアッド0,0,1,2(803b)に進行する。このことはさらに図10を参照して記述して行く。

【0029】

また図10について

・段階S4.1 サブクアッド803cに対応している点の*x*,*y*座標を読み込む；

・段階S4.2 サブクアッド0,0,1,2(803c)内のすべての点について取囲んでいる箱を製図して、“シュリンクラップ”したサブクアッド0,0,1,2(これは単一の点である)を作る；

・段階S4.3及び段階S4.4 この単一の点についての広がり(extent)すなわち*d*_{*x*},*d*_{*y*}をセーブし、またこの点の座標をセーブする；

・段階S4.5 この点が正の寸法であるかどうか、すなわちこの点の*d*_{*x*},*d*_{*y*}がゼロに等しいかどうかをチェックする。事實は、サブクアッド803cが単一の点に分解しているので*d*_{*x*}と*d*_{*y*}とは共にゼロである。そこで段階S4.8へ進む；

・段階S4.8 このレベル(0,0,1,*i*)でサブクアッドカウンタ*i*をINCREMENTする。サブクアッド0,0,1,3(803d)内部には点は存在しないので(段階S4.7)、そこで段階S4.8に戻り：サブクアッドカウンタ*i*をこのレベル(0,0,1,*i*)でINCREMENTする：しかし*i*>3であるから、

・段階S4.9 サブクアッドレベル0,0,*i*に戻って、サブクアッドカウンタを1から2へINCREMENTして、ここでサブクアッド0,0,2(703c)を考える：サブクアッド0,0,2(703c)には一つ点があり、したがってサブクアッド0,0,2(703c)内部にこの点を入れ(段階S4.7)て、段階S4.1へ向い、段階S4.1を介してサブクアッド0,0,2(703c)について進行する。このことは図11を参照して説明するところである。

【0030】

10

20

30

40

50

図 1 1

- ・段階 S 4 . 1 サブクアッド 7 0 3 c に対応している点の x , y 座標を読み込む ;
- ・段階 S 4 . 2 サブクアッド 0 , 0 , 2 (7 0 3 c) 内のすべての点について取囲んでいる箱を製図して、 “ シュリンクラップ ” したサブクアッド 0 , 0 , 2 (これは単一の点である) を創る。
- ・段階 S 4 . 3 及び段階 S 4 . 4 この単一の点の広がり の程度すなわち $d x$, $d y$ をセーブし、この点の座標をセーブする ;
- ・段階 S 4 . 5 この点が正の寸法をもっているかどうか、すなわちこの点の $d x$, $d y$ がゼロに等しいかどうかをチェックする。事實は、サブクアッド 7 0 3 c は単一の点にまで分解されているので $d x$ と $d y$ とは共にゼロである。したがって段階 S 4 . 8 となる ;
- ・段階 S 4 . 8 このレベル (0 , 0 , i) でサブクアッドカウンタ i をインCREMENTする。サブクアッド 0 , 0 , 3 (7 0 3 d) 内部には点は存在しない (段階 S 4 . 7) ので段階 S 4 . 8 へ戻る : このレベル (0 , 0 , i) でサブクアッドカウンタ i をインCREMENTする : しかし $i > 3$ であるからそこで
- ・段階 S 4 . 9 サブクアッドレベル 0 , i に戻り、サブクアッドカウンタ i を 0 から 1 へインCREMENTして、これによりサブクアッド 0 , 1 (6 0 3 b) を考慮する。このことが図 1 2 を参照して記述される。

【 0 0 3 1 】

図 1 2

図 7 ~ 11 で記述したプロセスが繰返されるが、サブクアッド 0 , 1 については、サブクアッド 0 , n 内部のすべての点が一度サブクアッドに指定されてしまうと、このプロセスはサブクアッド 1 に向けて移動することになる。

【 0 0 3 2 】

始めに記述したように、こういった点に向けてのインデックスを構築することは、サブクアッド情報をセーブすることという事項である。このことについてはたくさんのやり方が工夫考案されているが、好ましいのは、インデックスが段階 S 4 . 3 と S 4 . 4 とでセーブされたサブクアッド情報を備えていることであり、言い換えると、 x , y 座標でのサブクアッドの広がり の程度 (extent) と、その中に落ち込むことになる点の座標であり、またそのサブクアッド内部の四つのサブクォードラント (sub - quadrants) へのリンクであることとされる。したがって、このインデックスは本質的にはサブクアッド構造の階層構造を備え、この構造は階層構造が各継続するサブクアッドとその四つのサブクアッドとの間の関係によって与えられるものとなっている。サブクアッドはサブクアッド階層構造に従ってインデックスに書込まれ、このサブクアッド階層構造は最大のサブクアッド (ここでは図 6 の 5 0 1) から一番小さいサブクアッドに下っていくものである。

【 0 0 3 3 】

データベース DB 1 内にサブクアッド構造をセーブすることに加えて、点がこのデータベースに書込まれる。このデータベースは同じものでも異なるものでもよく、例えばファイル内にサブクアッド階層構造の反転したもので与えられる順序で書込まれる。したがって、この場合には、階層構造の底のサブクアッド内の点が先ずファイル内に書込まれる。点がファイルに書込まれるにつれて、点の全数についての移動書付け記録 (running tally) が維持されて、ファイルに各点が書かれると、カウンタが次のように提示する : “ current number of points encountered so far + 1 ” (これまでに出会った点の現在数プラス 1) であり、これがそれぞれのサブクアッド構造に書込まれる。移動書付け記録は一番小さなサブクアッドから上方へ作業をして行き、各サブクアッドについては、そのサブクアッド内のすべての点の第一のものの位置をインデックスを付けられているすべての点によって (Pos_{sub_quad}) 本質的に表示する。例えば、図 1 3 を参照すると、次のようになる。

【 0 0 3 4 】

【表 1】

10

20

30

40

サブクアド	サブクアド内の点 (N)	サブクアドに向けて 書かれた第一の点の数 (Pos _{sub_quad})	点ファイルの中でハイ ライトをあてた点
0, 0, 1, 1	n (N=1)	1	<u>n</u> , m, p, o...
0, 0, 1, 2	m (N=1)	2	n, <u>m</u> , p, o...
0, 0, 0	p (N=1)	3	n, m, <u>p</u> , o...
0, 0, 1	n, m (N=2)	1	<u>n</u> , m, p, o...
0, 0, 2	o (N=1)	4	n, m, p, <u>o</u> ...
0, 0	n, m, p, o (N=4)	1	<u>n</u> , m, p, o...
0, 1, c, d	若干の点	5	n, m, p, o, 次...

10

【 0 0 3 5 】

こうして、サブクアド 5 0 1 (一番外側のクアド、図 6 を参照) についての点のファイルは n, m, p, o... を読む (これが一番小さいサブクアド 0, 0, 1, 1 から始まっている)。点の数 (N) と、点のサブクアド内の第一の点に対応している移動書付け記録 (Pos_{sub_quad}) の数の両方がそのサブクアドに対して書かれるので、次に、関心のあるサブクアドが一旦識別されると、識別されたサブクアド内部に存在する点が点ファイル内で Pos_{sub_quad} へ移動してその位置から N 個の点を抽出することによって抽出ができる。このことが点の検索を示している実施例の中で示されている。

20

【 0 0 3 6 】

情報の検索 (Retrieval)

第二の発明は、各要素と他の要素でそのインデックス内のものとの間の関係と、インデックス内の要素とインデックスを付けられているエンティティとの間の関係とが既知のときに、エンティティと関係している要素のインデックスによってエンティティを検索する方法に関するものである。この方法は上で提示したインデックス付けの方法に従って創り出されたインデックスに対して直ちに適用ができるのであり、その理由は、このインデックスがサブクアド構造の階層構造を備えていて、またこの階層構造が (クアド > サブクアド関係により) よく定義されていることによる。しかしながら、この方法は、こういった条件を満足するいずれの形式のインデックスに対して等しく適用できることに留意されたい。

30

【 0 0 3 7 】

以下の記述では、検索プロセスの実施例が記述され、そこでは点に対するインデックスが問合せが “関心のある区域 (region of interest)” を特定する問合せで照合される。次のことが仮定される：

- ・インデックス内の要素は所定領域内部の複数の領域である；
- ・インデックスを付けられるエンティティは、二次元によって定義される点である；
- ・領域間には所定の関係が存在する；また
- ・領域と点との間には所定の関係が存在する。

40

本質的なことは、実施例が領域のどれが次のものであるかを識別するというのである：

- a) 関心のある区域と重なる、また
- b) 関心のある区域と重なるこういった領域内部に点を含んでいる。

領域と点との間の所定の関係がそこで使用されて識別された領域内部に落ち込む点を抽出する。

【 0 0 3 8 】

関心のある領域はエンティティの二次元提示内部の区域を指している。したがって、上述

50

の時間的なレンジ (range) (すなわち、営業やレジヤ施設の閉じたり開いたりする時間) については、関心のある区域は関心のある時間期間であって、“店は10時と13時の間開いている”といったものである。関心のある区域はそこである区域(好ましいのは正方形)によって定義されることで、この区域は前もって特定された点の内部に取囲まれている。図35a~eを参照すると、関心のある領域は次のいずれか一つとなることができる：

(0, 10), (13, 24) 10時と13時との間のある時間開いている機関(図35a)

(0, 13), (10, 24) 10時と13時との間連続して開いている機関(図35b)

(0, 10), (10, 13) 10時前に開くが13時前に閉じる機関(図33c)

(10, 13), (13, 24) 10時後に開くが13時後に閉じる機関(図33d)

(10, 10), (13, 13) 10時後に開きしかも13時前に閉じる機関(図33e)

同じように、上述した値段付け範囲については、関心のある区域は価格のレンジ(範囲)であり、それ故に、「範囲が英ポンド50.00ないし80.00の範囲のどこかに入ることになるすべてのアイテム」についての検索要件については、関心のある区域は点(0, 50)と(80, 80)との内部に限定された区域によって定義できる。

【0039】

上述した位置情報については、関心のある区域は場所の範囲であり、例えば「第一の場所((lat, long)₁)と第二の場所((lat, long)₂)の間に置かれた全アイテム」のようなものである。

【0040】

関心のある区域と重なる領域を識別する方法の段階を示している流れ図は、関心のある区域が位置情報と関係しているときに、図14と図15とに示されている。これらの段階は、図16に示した関心のある区域については、図16ないし図33で示されている。この実施例では、所定区域内部の領域の特定例がクアッドとサブクアッドと呼ばれている。図14, 図15に示した方法の段階は以下に図16ないし図33の各々を参照して記述されている。

【0041】

図16

・S13.1.1 関心のある区域(x₁, y₁)(x₂, y₂)のx, y座標を読み込む。例をあげると、もしユーザがある領域内のガレージ(garages)を位置決めしたいとすると、これが緯度/経度の対としてインデックス付けされて、二次元の空間内の点を定義するものとなり、またユーザが関心をもっている地理的領域が、その二次元の空間内で「関心のある区域」として表現されることができる。

・S13.1.2 点の座標と一番外側のクアッドについてのクアッドの寸法とを検索し、一番外側のクアッドの寸法に対して(X₁__Q, Y₁__Q), (X₂__Q, Y₂__Q)を設定する；

・S13.2 関心のある区域が切り取り(cropping)を必要とするかどうか評価する：もしx₁ < X₁__Qであればx₁ = X₁__Qと設定し；もしy₁ < Y₁__Qであればy₁ = Y₁__Qと設定し；もしx₂ > X₂__Qであればx₂ = X₂__Qと設定し；もしy₂ > Y₂__Qであればy₂ = Y₂__Qと設定する。例としての関心のある区域で図16に示したものは、これらの条件のいずれも満足していないと；

・S13.3 x₁ > x₂もしくはy₁ > y₂かどうかを評価する。この場合、いずれの条件も満足されていないと；

段階S13.2とS13.3とは条件の唯一の例であり、この条件は関心のある区域とS13.1.2で検索されたサブクアッドが重なるかどうか、また、もし重なりが存在すれば重なっている区域内にいずれかの点が存在するかどうかを確定するために適用できるものである；この確定に適用するために代りの方法が適用できるものと考えている。

10

20

30

40

50

・ S 1 3 . 4 関心のある区域がそのクアッドと正確に重なっているかどうか評価する。
否 (N O) であれば :

・ S 1 3 . 5 図 1 7 を参照して記述するように、 S 1 3 . 1 . 2 に従って現在のクアッド (5 0 1) のサブクアッド (5 0 3 a) を検索する。

【 0 0 4 2 】

図 1 7

・ S 1 3 . 1 . 2 “ シュリンクラップ ” したサブクアッド 0 (6 0 1) から点の座標とサブクアッドの寸法とを検索する : “ シュリンクラップ ” したサブクアッド 6 0 1 の寸法に、 (X 1 _ Q , Y 1 _ Q) (X 2 _ Q , Y 2 _ Q) を設定する ;

・ S 1 3 . 2 関心のある区域が切取り (cropping) を必要としているかどうか評価する : もし $x_1 < X_1_Q$ であれば $x_1 = X_1_Q$ と設定し ; もし $y_1 < Y_1_Q$ であれば $y_1 = Y_1_Q$ と設定し ; もし $x_2 > X_2_Q$ であれば $x_2 = X_2_Q$ と設定し ; もし $y_2 > Y_2_Q$ であれば $y_2 = Y_2_Q$ と設定する。この場合こういった条件がいずれも満足されていないと ;

・ S 1 3 . 3 $x_1 > x_2$ または $y_1 > y_2$ かどうか評価する。この場合にいずれの条件も満足されていないと ;

・ S 1 3 . 4 関心のある区域がそのクアッドと正確に重なっているかどうか評価する。
否であれば :

・ S 1 3 . 5 図 1 8 と 1 9 を参照して記述するように、 S 1 3 . 1 . 2 に従って現在のサブクアッドのサブクアッドを検索する。

【 0 0 4 3 】

図 1 8 及び図 1 9

・ S 1 3 . 1 . 2 “ シュリンクラップ ” したサブクアッド 0 , 0 (7 0 1) から点の座標とサブクアッドの寸法とを検索する : (X 1 _ Q , Y 1 _ Q) (X 2 _ Q , Y 2 _ Q) を “ シュリンクラップ ” したサブクアッド 7 0 1 の寸法に設定する ;

・ S 1 3 . 2 関心のある区域が切取り (cropping) を必要としているかどうか評価する : $x_2 > X_2_Q$ であれば $x_2 = X_2_Q$ と設定し ; また $y_2 > Y_2_Q$ であれば $y_2 = Y_2_Q$ と設定する ;

・ S 1 3 . 3 $x_1 > x_2$ あるいは $y_1 > y_2$ かどうかを評価する。この場合 (図 1 9) 両条件が満足されているならば、これが関心のある区域内部に落ち込むサブクアッド 0 , 0 (7 0 1) 内の点が存在しないことを意味する ;

・ S 1 3 . 6 このレベル (0 , i) でサブクアッドカウンタ i をインクリメントし、図 2 0 と 2 1 を参照して記述するように S 1 3 . 1 . 2 に従ってサブクアッド (0 , 1) を検索する。

【 0 0 4 4 】

図 2 0 及び図 2 1

・ S 1 3 . 1 . 2 “ シュリンクラップ ” したサブクアッド 0 , 1 から、点の座標とサブクアッドの寸法とを検索する : (X 1 _ Q , Y 1 _ Q) (X 2 _ Q , Y 2 _ Q) を “ シュリンクラップ ” したサブクアッド 0 , 1 の寸法に対して設定する ;

・ S 1 3 . 2 関心のある区域が切取り (cropping) を必要としているかどうか評価する : $x_1 < X_1_Q$ であれば $x_1 = X_1_Q$ と設定し、また $y_2 > Y_2_Q$ であれば $y_2 = Y_2_Q$ と設定する ;

・ S 1 3 . 3 $x_1 > x_2$ もしくは $y_1 > y_2$ であるかどうか評価する。いずれの条件も満足されていない場合には ;

・ S 1 3 . 4 関心のある区域 (今では切取りがされている) がサブクアッドと正確に重なっているかどうか評価する。否であれば :

・ S 1 3 . 5 図 2 2 (a) と図 2 2 (b) とを参照して記述するように、 S 1 3 . 1 . 2 に従って現在のサブクアッド 0 , 1 のサブクアッド 0 , 1 , 0 を検索する。

【 0 0 4 5 】

図 2 2 (a) 及び図 2 2 (b)

10

20

30

40

50

- ・ S 1 3 . 1 . 2 “シュリンクラップ”したサブクアッド 0 , 1 , 0 から、点の座標とサブクアッドの寸法を検索する： (X 1 __ Q , Y 1 __ Q) (X 2 __ Q , Y 2 __ Q) を “シュリンクラップ”したサブクアッド 0 , 1 , 0 の寸法に設定する；
- ・ S 1 3 . 2 関心のある区域が切取り (cropping) を必要とするかどうか評価する： $x_2 > X_2_Q$ であれば $x_2 = X_2_Q$ と設定し、また $y_2 > Y_2_Q$ であれば $y_2 = Y_2_Q$ と設定する；
- ・ S 1 3 . 3 $x_1 > x_2$ あるいは $y_1 > y_2$ かどうか評価する。この場合 (図 2 2 b) 両条件が満足されたならば、すなわち関心のある区域内に入るサブクアッド 0 , 1 , 0 内に点が存在していないことを意味する；
- ・ S 1 3 . 6 このレベル (0 , 1 , i) でサブクアッドカウンタ i をインCREMENTし、図 2 3 (a) , 2 3 図 (b) を参照して記述するように、S 1 3 . 1 . 2 に従ってサブクアッド (0 , 1 , 1) を検索する (S 1 3 . 6 . 1) 。

10

【 0 0 4 6 】

図 2 3 (a) 及び 2 3 図 (b)

- ・ S 1 3 . 1 . 2 “シュリンクラップ”したサブクアッド 0 , 1 , 1 から点の座標とサブクアッドの寸法とを検索する：“シュリンクラップ”したサブクアッド 0 , 1 , 1 (すなわち単一の点) の寸法に (X 1 __ Q , Y 1 __ Q) (X 2 __ Q , Y 2 __ Q) を設定する。
- ・ S 1 3 . 2 関心のある区域が切取り (cropping) を必要としているかどうか評価する： $x_1 < X_1_Q$ であれば $x_1 = X_1_Q$ と設定し、 $y_2 > Y_2_Q$ であれば $y_2 = Y_2_Q$ と設定する；
- ・ S 1 3 . 3 $x_1 > x_2$ あるいは $y_1 > y_2$ であるかどうか評価する。この場合 (図 2 3 (b)) に両方の条件が満足されれば、関心のある区域内に入るサブクアッド 0 , 1 , 1 内に点が存在しないことを意味する；
- ・ S 1 3 . 6 このレベル (0 , 1 , i) でサブクアッドカウンタ i をインCREMENTし、S 1 3 . 1 . 2 に従ってサブクアッド (0 , 1 , 2) を検索する (S 1 3 . 6 . 1) 。このことは図 2 4 (a) と 2 4 (b) とを参照して記述するところである。

20

【 0 0 4 7 】

図 2 4 (a) 及び 図 2 4 (b)

- ・ S 1 3 . 1 . 2 “シュリンクラップ”したサブクアッド 0 , 1 , 2 から点の座標とサブクアッドの寸法とを検索する： (X 1 __ Q , Y 1 __ Q) (X 2 __ Q , Y 2 __ Q) を “シュリンクラップ”したサブクアッド 0 , 1 , 2 すなわち単一の点に設定する；
- ・ S 1 3 . 2 関心のある区域が切取り (cropping) を要求しているかどうか評価する： $x_1 < X_1_Q$ であれば $x_1 = X_1_Q$ と設定し、 $y_1 < Y_1_Q$, $x_2 > X_2_Q$ 及び $y_2 > Y_2_Q$ であれば $x_1 = X_1_Q$, $y_1 = Y_1_Q$, $x_2 = X_2_Q$, $y_2 = Y_2_Q$ と設定する。
- ・ S 1 3 . 3 $x_1 > x_2$ もしくは $y_1 > y_2$ であるかどうか評価する。否であれば
- ・ S 1 3 . 4 関心のある区域 (今までは切取られている) がサブクアッドと正確に重なるかどうか評価する：肯定 (Y E S) であれば
- ・ S 1 3 . 4 . 2 サブクアッドとそのサブクアッド内の点の数 (ここでは 1) とを記録する；
- ・ S 1 3 . 6 このレベル (0 , 1 , i) でサブクアッドカウンタ i をインCREMENTし、図 2 5 (a) 及び 2 5 図 (b) を参照して記述するように S 1 3 . 1 . 2 に従ってサブクアッド (0 , 1 , 3) を検索する (S 1 3 . 6 . 1) 。

30

40

【 0 0 4 8 】

図 2 5 (a) 及び 図 2 5 (b)

- ・ S 1 3 . 1 . 2 “シュリンクラップ”したサブクアッド 0 , 1 , 3 から点の座標とサブクアッドの寸法とを検索する： (X 1 __ Q , Y 1 __ Q) (X 2 __ Q , Y 2 __ Q) を “シュリンクラップ”したサブクアッド 0 , 1 , 3 すなわち単一の点に対して設定する；
- ・ S 1 3 . 2 関心のある区域が切取り (クロッピング) を必要としているかどうか評価する： $x_1 < X_1_Q$ であれば $x_1 = X_1_Q$ と設定し、 $y_1 < Y_1_Q$, $x_2 > X_2_Q$ 及び $y_2 > Y_2_Q$ であれば $x_1 = X_1_Q$, $y_1 = Y_1_Q$, $x_2 = X_2_Q$, $y_2 = Y_2_Q$ と設定する；

50

Q及び $y_2 > Y_2_Q$ であれば $x_1 = X_1_Q$, $y_1 = Y_1_Q$, $x_2 = X_2_Q$, $y_2 = Y_2_Q$ と設定する;

- ・ S 1 3 . 3 $x_1 > x_2$ もしくは $y_1 > y_2$ であるかどうか評価する。否であると
- ・ S 1 3 . 4 関心のある区域(今では切取られている)がサブクアッドと正確に重なるかどうか評価する:肯定(Y E S)であれば
- ・ S 1 3 . 4 . 2 サブクアッドとサブクアッド内の点の数(ここでは1)とを記録する;
- ・ S 1 3 . 6 このレベル(0, 1, i) ... $i > 3$ でサブクアッドカウンタiをインクリメントし、図26, 27を参照して記述するようにS 1 3 . 1 . 2に従ってサブクアッド(0, 2)を検索する(S 1 3 . 6 . 2)。

10

【0049】

図26及び図27

- ・ S 1 3 . 1 . 2 “シュリンクラップ”したサブクアッド0, 2から、点の座標とサブクアッドの寸法とを検索する: (X_1_Q, Y_1_Q) (X_2_Q, Y_2_Q) を“シュリンクラップ”したサブクアッド0, 2の寸法に対して設定する;
- ・ S 1 3 . 2 関心のある区域が切取り(クロッピング)を要求するかどうかを評価する: $y_1 < Y_1_Q$ であれば $y_1 = Y_1_Q$ と設定し、また $x_2 > X_2_Q$ であれば $x_2 = X_2_Q$ と設定する。
- ・ S 1 3 . 3 $x_1 > x_2$ もしくは $y_1 > y_2$ であるかどうか評価する。この場合にいずれの条件も満足されていなければ;
- ・ S 1 3 . 4 関心のある区域(今では切取りがされている)がサブクアッドと正確に重なるかどうか評価する。否であると;
- ・ S 1 3 . 5 現在のサブクアッド0, 2のサブクアッド0, 2, 0をS 1 3 . 1 . 2に従って、図28(a)と図28(b)とを参照して記述するように検索し出す。

20

【0050】

図28(a)及び図28(b)

- ・ S 1 3 . 1 . 2 “シュリンクラップ”したサブクアッド0, 2, 0から点の座標とサブクアッドの寸法とを検索する: (X_1_Q, Y_1_Q) (X_2_Q, Y_2_Q) を“シュリンクラップ”したサブクアッド0, 2, 0の寸法に対して設定する;
- ・ S 1 3 . 2 関心のある区域が切取り(クロッピング)を必要としているかどうか評価する: $x_2 > X_2_Q$ であれば $x_2 = X_2_Q$ と設定し、また $y_2 > Y_2_Q$ であれば $y_2 = Y_2_Q$ と設定する;
- ・ S 1 3 . 3 $x_1 > x_2$ もしくは $y_1 > y_2$ かどうかを評価する。この場合(図28(b))、両方の条件が満足されると、関心のある区域内部に入るサブクアッド0, 2, 0内に点が存在しないことを意味する;
- ・ S 1 3 . 6 このレベル(0, 2, i)でサブクアッドカウンタiをインクリメントし、図29(a)と図29(b)とを参照して記述するように、S 1 3 . 1 . 2に従ってサブクアッド(0, 2, 1)を検索する(S 1 3 . 6 . 1)。

30

【0051】

図29(a)及び図29(b)

- ・ S 1 3 . 1 . 2 “シュリンクラップ”したサブクアッド0, 2, 1から点の座標とサブクアッドの寸法とを検索する: (X_1_Q, Y_1_Q) (X_2_Q, Y_2_Q) を“シュリンクラップ”したサブクアッド0, 2, 1すなわち単一の点の寸法に設定する;
- ・ S 1 3 . 2 関心のある区域が切取り(クロッピング)を必要としているかどうか評価する: $x_1 < X_1_Q$ であれば $x_1 = X_1_Q$ と設定し、 $y_1 < Y_1_Q$, $x_2 = X_2_Q$ 及び $y_2 > Y_2_Q$ であれば $x_1 = X_1_Q$, $y_1 = Y_1_Q$, $x_2 = X_2_Q$, $y_2 = Y_2_Q$ と設定する;
- ・ S 1 3 . 3 $x_1 > x_2$ 或いは $y_1 > y_2$ かどうか評価する。否であれば
- ・ S 1 3 . 4 関心のある区域(今では切取られている)がサブクアッドと正確に重なるかどうか評価する:肯定であれば

40

50

・ S 1 3 . 4 . 2 サブクアッドとサブクアッド内部の点の数（ここでは 1）とを記録する；

・ S 1 3 . 6 このレベル (0 , 1 , i) ... $i < 3$ でサブクアッドカウンタ i をインクリメントして、図 3 0 (a) と図 3 0 (b) とを参照して記述するように S 1 3 . 1 . 2 に従ってサブクアッド (0 , 2 , 2) を検索する (S 1 3 . 6 . 1) 。

【 0 0 5 2 】

図 3 0 (a) 及び図 3 0 (b)

・ S 1 3 . 1 . 2 “シュリンクラップ”したサブクアッド 0 , 2 , 2 から点の座標とサブクアッドの寸法とを検索する： ($X 1 _ Q$, $Y 1 _ Q$) ($X 2 _ Q$, $Y 2 _ Q$) を “シュリンクラップ”したサブクアッド 0 , 2 , 2 のサイズに対して設定する；

・ S 1 3 . 2 関心のある区域が切取り (クロッピング) を必要としているかどうか評価する： $x 2 > X 2 _ Q$ であれば $x 2 = X 2 _ Q$ と設定し、また $y 1 < Y 1 _ Q$ であれば $y 1 = Y 1 _ Q$ と設定する；

・ S 1 3 . 3 $x 1 > x 2$ あるいは $y 1 > y 2$ かどうか評価する。この場合 (図 3 0 (b)) 両方の条件が満足されていれば、関心のある区域内に落ち込むサブクアッド 0 , 2 , 2 内の点は存在しないことを意味する；

・ S 1 3 . 6 このレベル (0 , 2 , i) でサブクアッドカウンタ i をインクリメントし、また図 3 1 を参照して記述するように、S 1 3 . 1 . 2 に従ってサブクアッド (0 , 2 , 3) を検索する (S 1 3 . 6 . 1) 。

【 0 0 5 3 】

図 3 1

・ S 1 3 . 1 . 2 “シュリンクラップ”したサブクアッド 0 , 2 , 3 から点の座標とサブクアッドの寸法とを検索する： 0 , 2 , 3 に対応しているサブクアッドは存在しないがその理由はこのサブクアッドに対応している領域内に点が存在しないことであり、そこで S 1 3 . 6 へ跳び

・ S 1 3 . 6 . 2 このレベル (0 , 2 , i) : $i > 3$ でサブクアッドカウンタ i をインクリメントして、それにより図 3 2 及び図 3 3 を参照して記述するように S 1 3 . 1 . 2 に従ってサブクアッド (0 , 3) を検索する (S 1 3 . 6 . 2) 。

【 0 0 5 4 】

図 3 2 及び図 3 3

・ S 1 3 . 2 関心のある区域が切取り (クロッピング) を必要とするかどうか評価する： $x 1 < X 1 _ Q$ であれば $x 1 = X 1 _ Q$ と設定し、 $y 1 < Y 1 _ Q$, $x 2 > X 2 _ Q$ 及び $y 2 > Y 2 _ Q$ であれば $x 1 = X 1 _ Q$, $y 1 = Y 1 _ Q$, $x 2 = X 2 _ Q$, $y 2 = Y 2 _ Q$ と設定する。

・ S 1 3 . 3 $x 1 > x 2$ か $y 1 > y 2$ かどうかを評価する： 否であれば

・ S 1 3 . 4 関心のある区域 (今では切取られている) が正確にサブクアッドと重なるかどうか評価する： 肯定であれば

・ S 1 3 . 4 . 2 サブクアッドとサブクアッド内部の点の数 (ここでは 2) を記録する；

・ S 1 3 . 6 このレベル (0 , i) ... $i > 3$ でサブクアッドカウンタ i をインクリメントし、それにより S 1 3 . 1 . 2 に従ってサブクアッド 1 を検索する (S 1 3 . 6 . 2)

【 0 0 5 5 】

図 1 6 から見るように、関心のある区域はサブクアッド 0 とその中にあるサブクアッドの内部にだけ入ることになり、それにより、図 1 4 と図 1 5 とに記述したプロセスがサブクアッド 1 , 2 , 3 に適用されるときには、段階 S 1 3 . 2 と S 1 3 . 3 で適用された条件 2 ~ 3 段階 (a few steps) 以内で該プロセスを終結する。

【 0 0 5 6 】

関心のある区域と重なっているサブクアッドを識別するためのプロセスの終わりでは、サブクアッドと、こういったサブクアッド内部の点の数であって段階 S 1 3 . 4 . 2 で記録さ

10

20

30

40

50

れたものが復帰される。この例は：

サブクアッド 0, 1, 2, 点の数：1 点

サブクアッド 0, 1, 3, 点の数：1 点

サブクアッド 0, 2, 1, 点の数：1 点

サブクアッド 0, 3, 点の数：2 点

ひとたびサブクアッドが識別されると、実際の点が検索される。この実施例では、上述したように、点はフラットファイル内に記憶されている。さらに、各サブクアッド構造は位置を示している数を記憶していて、それぞれのサブクアッド内部の第一の点についての、インデックスを付けられた点の全数と関係している（例えば図 4, 図 5 の例を参照するとクアッド 5 0 1 内部の点の全部の数と関係している）。点を実際に検索するためのプロセスは図 3 4 に示されている：

段階 S 1 3 . 4 . 2 で記録された各クアッドについては：

- ・ S 3 2 . 1 サブクアッド (N) の中に入る点の数をそのサブクアッドについて検索する；
- ・ S 3 2 . 2 対応しているサブクアッド構造 (Pos_{sub_quad}) から第一の点の位置を検索する；
- ・ S 3 2 . 3 Pos_{sub_quad} によって与えられる点のファイル内の位置にファイルポインタを移動する；
- ・ S 3 2 . 4 この位置から、ファイルから N 個の点を抽出する。

【 0 0 5 7 】

実施

図 4 及び図 5、図 1 4 及び図 1 5、また図 3 4 で記述したプロセスは、ソフトウェアで実施され、端末 T 3, T 4 の 1 つで実行されるか、端末 T 3, T 4 の間で分散される。端末 T 3, T 4 はしたがって複数の電算機の一つまたは複数を代表していて、好ましくはサーバコンピュータである。

【 0 0 5 8 】

インデックスを付けられることになる点は端末 T 3, T 4 にファイルもしくは類似のものを介して入力されることができ、上述したように創られたインデックスがデータベース DB 1 内に記憶されるようにでき、また点のファイルもデータベース DB 1 内に記憶されるようにできる。関心のある領域はデータベース問合せの形式で入力されることができ、クライアント端末 (図示せず) を介して入力され、既知のやり方で網 N 1 上で通信される。

【 0 0 5 9 】

好ましくは、上述のプロセスが C プログラム用言語で実施され、また反復的なプログラミング方法を使用して “バロウダウン (burrow down)” がサブクアッド内部のサブクアッドに対して行なわれることである。このような方法はこの発明にとって本質的なことではないと理解されたい。

【 0 0 6 0 】

追加的な詳記事項及び変形

上述したように、この発明は二次元で表現されたデータにインデックス付けをし検索するために使用できる。この発明はまた、二次元よりも高次元のデータをインデックス付けして検索するためにも使用できるのであって、 $n > 2$ である n 次元データが二次元に変換できるようにする。この場合に、変換された、二次元データはこの発明によってインデックスが付けられて検索される。例えば、三次元空間内で定義されたオブジェクトは N C A R グラヒックスのようなパッケージを用いる二次元に変換されることができ、N C A R グラヒックスは Unix に基づくグラフィックスパッケージであって、これが広範囲にわたる能力を数値データについての表示と処理とについて提供していて、大気研究用の大学共同体 (the University Corporation for Atmospheric Research) によって開発されたものである。(これに関して、<http://www.dkrz.de/ngdoc/ng4.0.1> を N C A R グラヒックスに関する情報について参照し、また、<http://ngwww.ucar.edu/ngdoc/ng/fund/chp16-21/threed.html> を 3 次元から 2 次元変換という問題に係る情報について参照されたい。)

10

20

30

40

50

その他の変形修正も可能であり、例えば、簡単なものにクアッドとサブクアッドとを各反復で異なる数の領域へ分割することをを用いることがあり、4に代って8とか10とすることが挙げられる。

【図面の簡単な説明】

【図1】 この発明により使用される通信システムの特徴を示す模式図。

【図2】 この発明によりインデックスを付けされる点（ポイント）の例を示す模式図。

【図3】 図2の拡大図を示す模式図。

【図4】 図4（4a）は、図5（4b）と組合わせて流れ図を構成し、図2で示した点にインデックス付けをするときの、この発明によるインデックス付けプロセスの実施例を示す。

10

【図5】 図5（4b）は、図4（4a）と組合わせて流れ図を構成し、図2で示した点にインデックス付けをするときの、この発明によるインデックス付けプロセスの実施例を示す。

【図6】 図2に示した点の周りにクアッドを創るための図4と図5のプロセスの適用を示す模式図。

【図7】 図6により創られたクアッドのサブクアッドを創るための図4と図5とのプロセスの適用を示す模式図。

【図8】 図7のサブクアッドの一つのサブクアッドを創るための図4と図5とのプロセスの適用を示す模式図。

【図9】 図8の拡大図。

20

【図10】 図8に示したサブクアッドとは別なものを創るための図4と図5とのプロセスの適用を示す図9の拡大図。

【図11】 図8に示したサブクアッドとは別なものを創るための図4と図5とのプロセスの適用を示す図9の拡大図。

【図12】 図7のサブクアッドの一つの別のサブクアッドを創るための図4と図5とのプロセスの適用を示す模式図。

【図13】 この発明による点を記憶するプロセスを示す模式図。

【図14】 図14（13a）は図15（13b）との組合わせにより流れ図を構成し、関心のある領域に従って点を検索するときのこの発明による検索プロセスの実施形態を示す。

30

【図15】 図15（13b）は図14（13a）との組合わせにより流れ図を構成し、関心のある領域に従って点を検索するときのこの発明による検索プロセスの実施形態を示す。

【図16】 点が検索されることになる関心のある領域の例を示す模式図。

【図17】 図7のサブクアッドの一つに対する図14と図15のプロセスの適用を示す模式図。

【図18】 図17の拡大図であり、図8のサブクアッドの第一のものに対して図14と図15のプロセスの適用を示す模式図。

【図19】 図17の拡大図であり、図8のサブクアッドの第一のものに対して図14と図15のプロセスの適用を示す模式図。

40

【図20】 図17の拡大図であり、関心のある領域と図8のサブクアッドの第二のものとのを示す図。

【図21】 図20に対応している模式図であり、図8のサブクアッドの第二のものに対して図14と図15のプロセスの適用を示す図。

【図22】 （a）と（b）とは図20に示したサブクアッドの（第一の）サブクアッドに対して図14と図15のプロセスの適用を示す模式図。

【図23】 （a）と（b）とは図20に示したサブクアッドの別の（第二の）サブクアッドに対して図14と図15のプロセスの適用を示す模式図。

【図24】 （a）と（b）とは図20に示したサブクアッドの別の（第三の）サブクアッドに対して図14と図15のプロセスの適用を示す模式図。

50

【図25】 (a)と(b)とは図20に示したサブアッドの別の(第四の)サブアッドに対して図14と図15のプロセスの適用を示す模式図。

【図26】 図17の拡大図であり、関心のある領域と、図8のサブアッドの第三のものを示す図。

【図27】 図26に対応している模式図であって、図8のサブアッドの第三のものに対して図14と図15のプロセスの適用を示す図。

【図28】 (a)と(b)とは図26に示したサブアッドの(第一の)サブアッドに対して図14と図15のプロセスの適用を示す模式図。

【図29】 (a)と(b)とは図26に示したサブアッドの別の(第二の)サブアッドに対して図14と図15のプロセスの適用を示す模式図。

【図30】 (a)と(b)とは図26に示したサブアッドの別の(第三の)サブアッドに対して図14と図15のプロセスの適用を示す模式図。

【図31】 (a)と(b)とは図26に示したサブアッドの別の(第四の)サブアッドに対して図14と図15のプロセスの適用を示す模式図。

【図32】 図17の拡大図であり、関心のある領域と図8のサブアッドの第四のものを示す図。

【図33】 図32に対応している模式図であり、図8のサブアッドの第四のものに対して図14と図15のプロセスの適用を示す図。

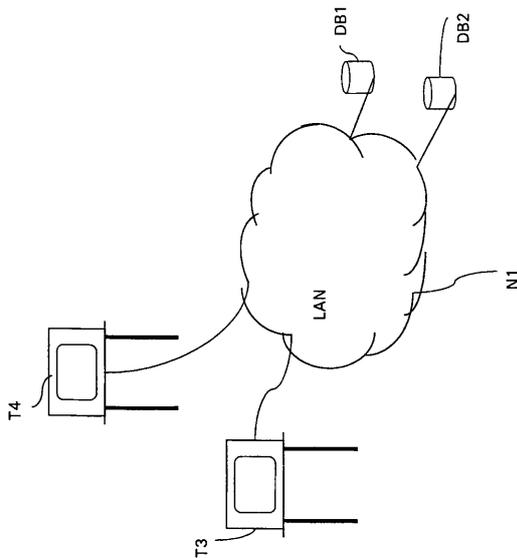
【図34】 図14と図15の検索プロセスに含まれるさらなるステップを示す流れ図。

【図35】 (a)ないし(e)の各図は関心のある領域についての模式的な例示であり、関心のある領域についてこの発明により点が検索されることになる。

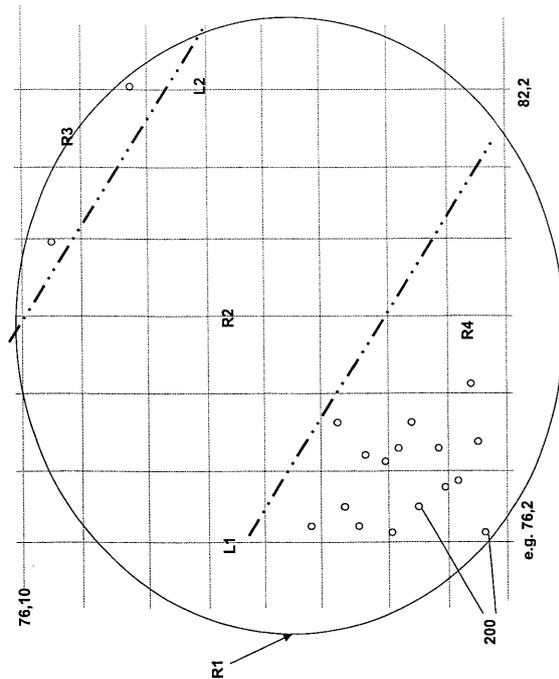
10

20

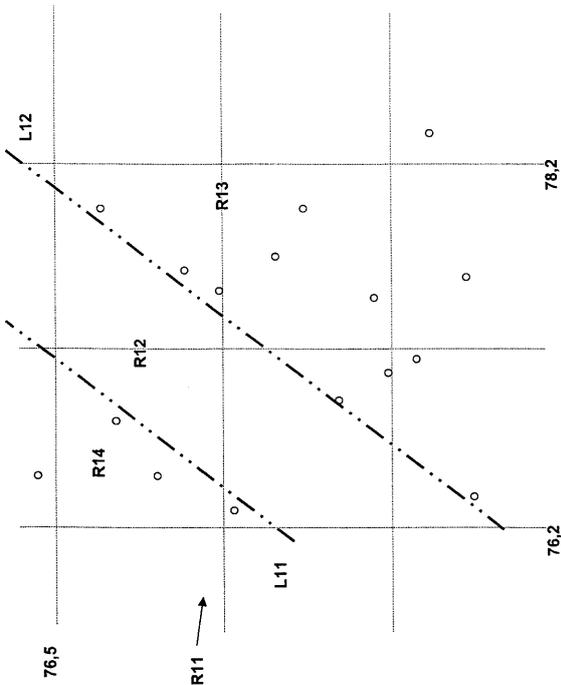
【図1】



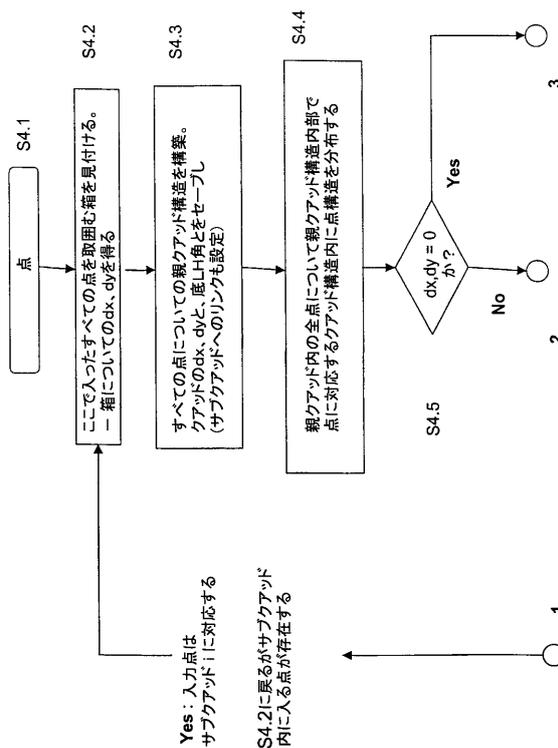
【図2】



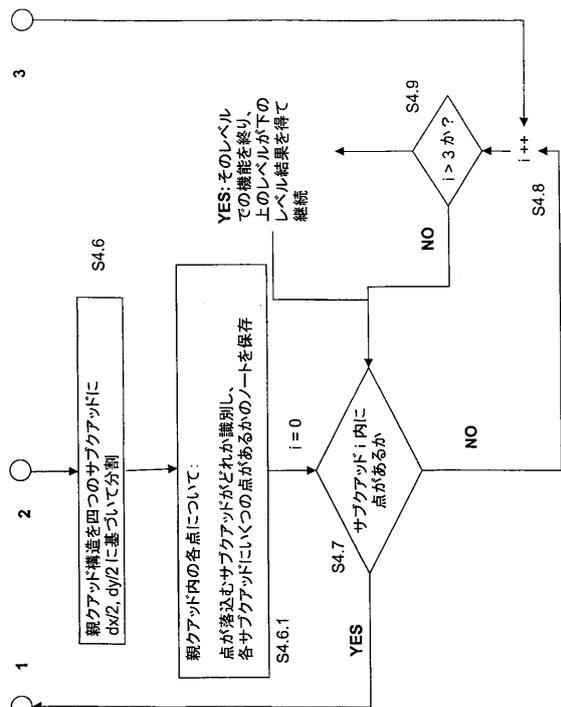
【 図 3 】



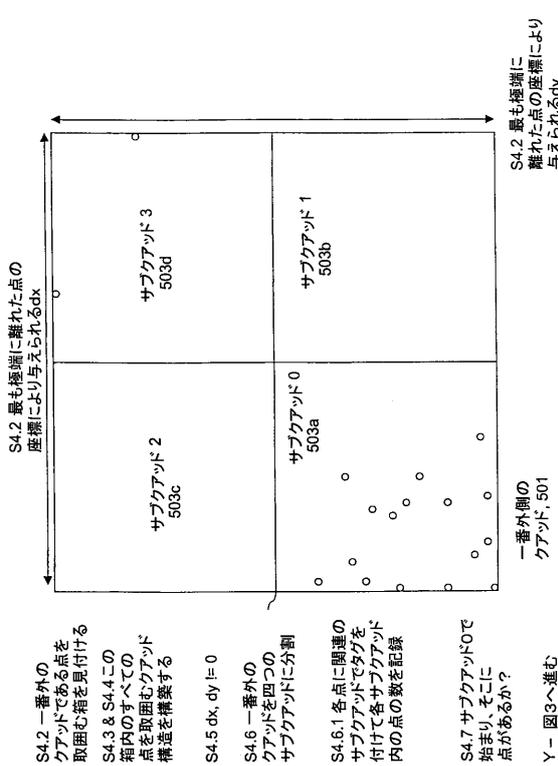
【 図 4 】



【 図 5 】



【 図 6 】



S4.2 一番外のクアッドである点を取囲む箱を見付ける

S4.3 & S4.4 この箱内のすべての点を取囲むクアッド構造を構築する

S4.5 dx, dy != 0

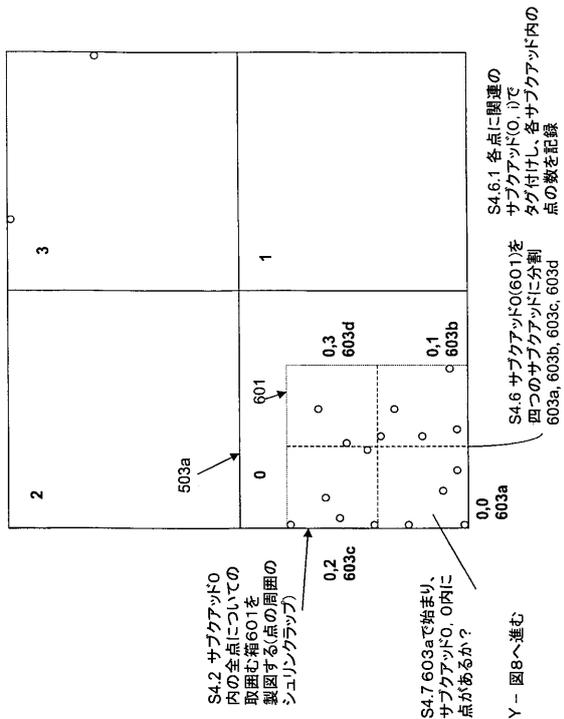
S4.6 一番外のクアッドを四つのサブクアッドに分割

S4.6.1 各点に関連のサブクアッドでタグを付けて各サブクアッド内の点の数を記録

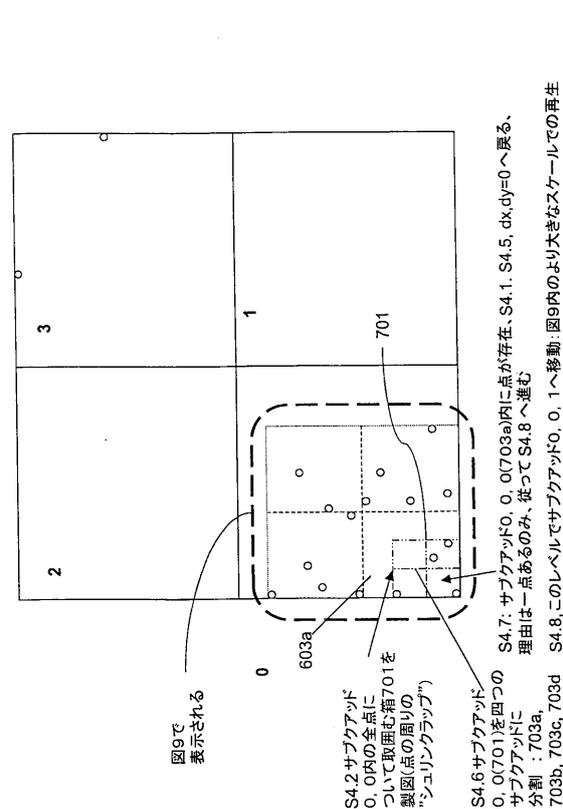
S4.7 サブクアッド0で始まり、そこに点があるか?

Y - 図3へ進む

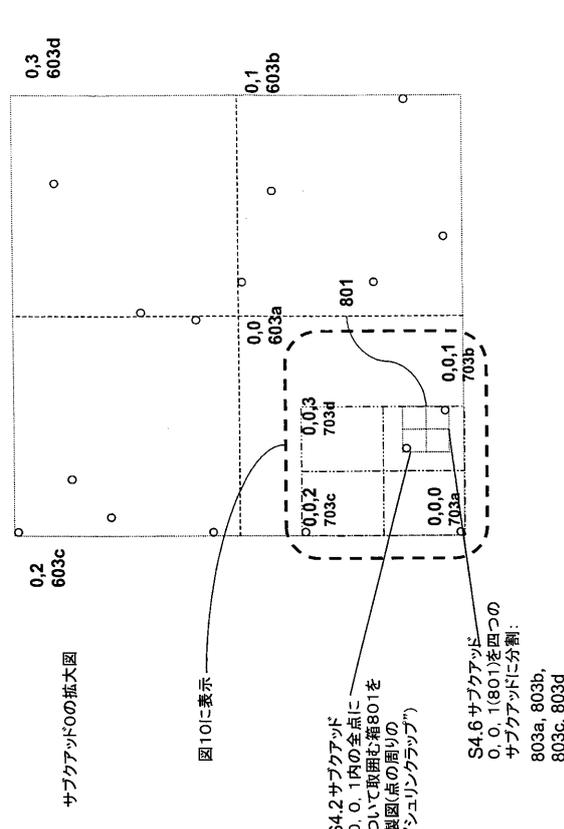
【 図 7 】



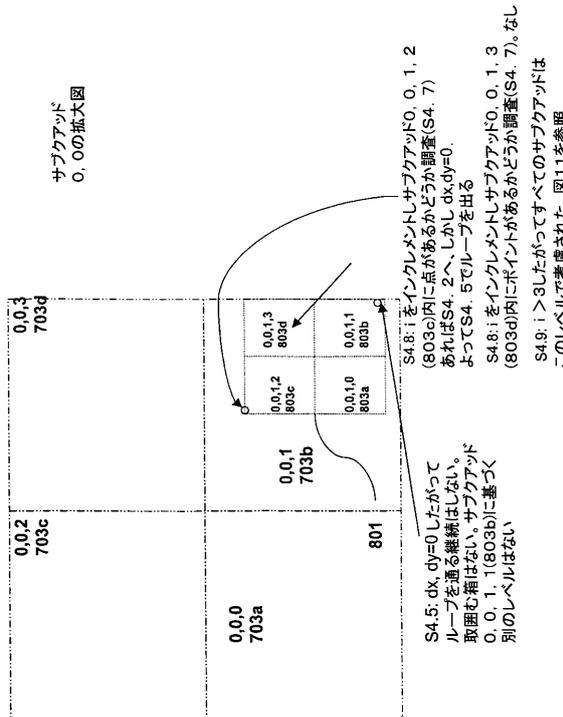
【 図 8 】



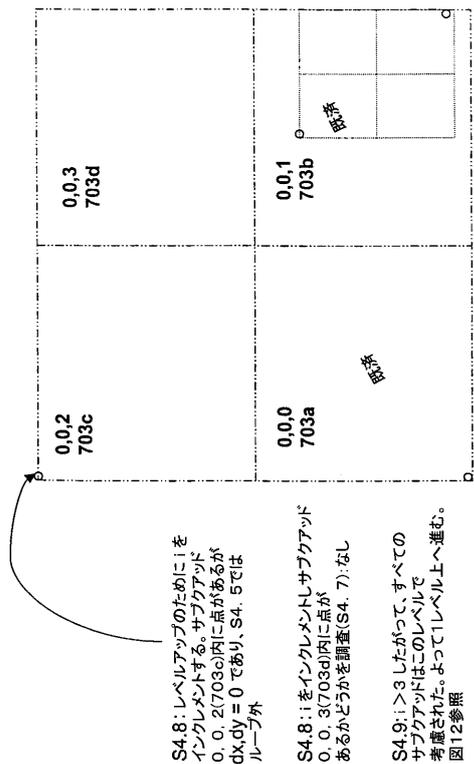
【 図 9 】



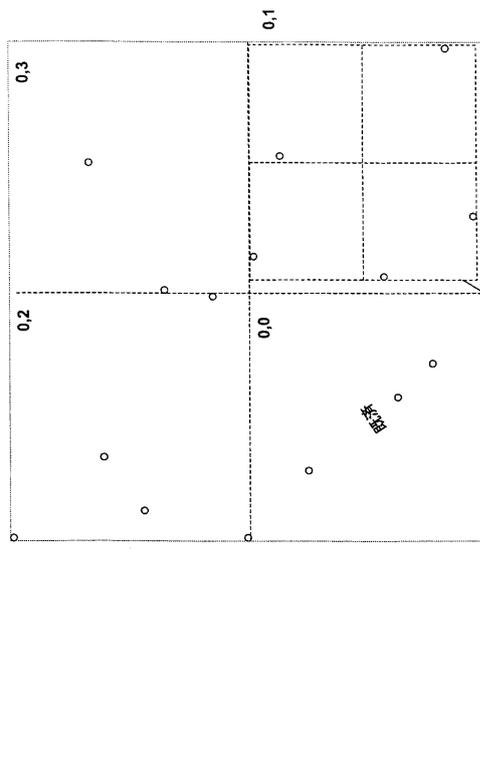
【 図 10 】



【 図 1 1 】

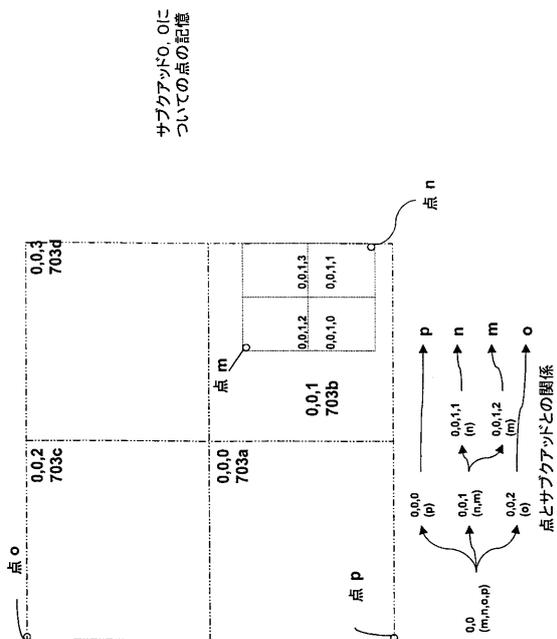


【 図 1 2 】

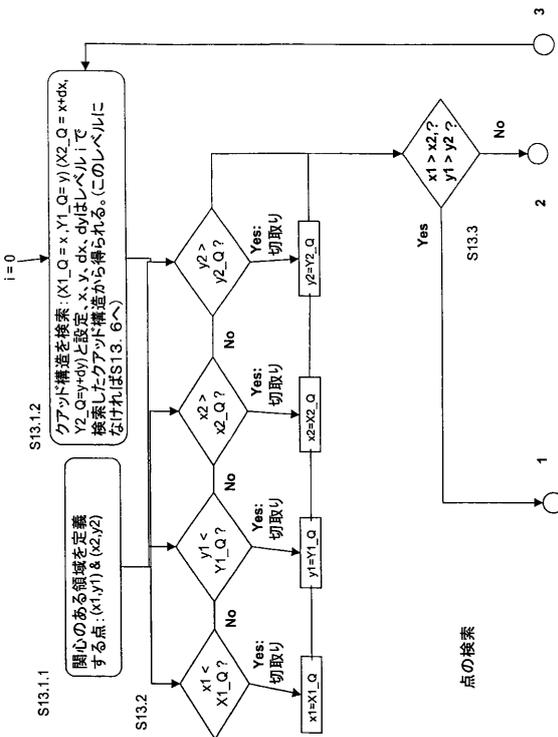


S4.2 サブアクト内の全体についての取囲む箱を製図
 図7~図11で記述したプロセスが繰返されるが、サブアクト0, 1について
 サブアクト0, n内部の全点が済んだら、プロセスはサブアクト1, nへ、以下続く

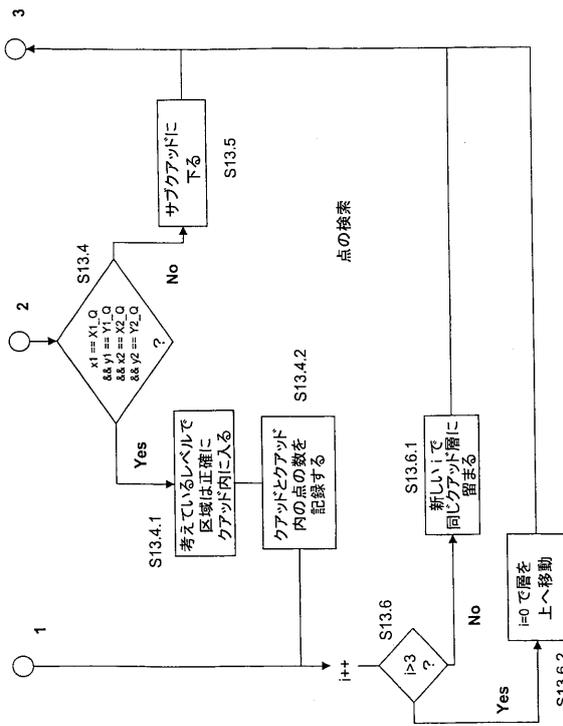
【 図 1 3 】



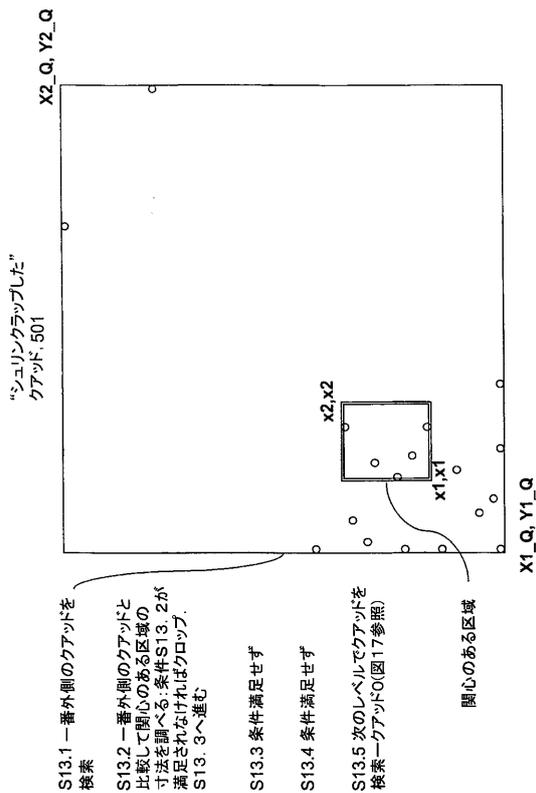
【 図 1 4 】



【 図 15 】

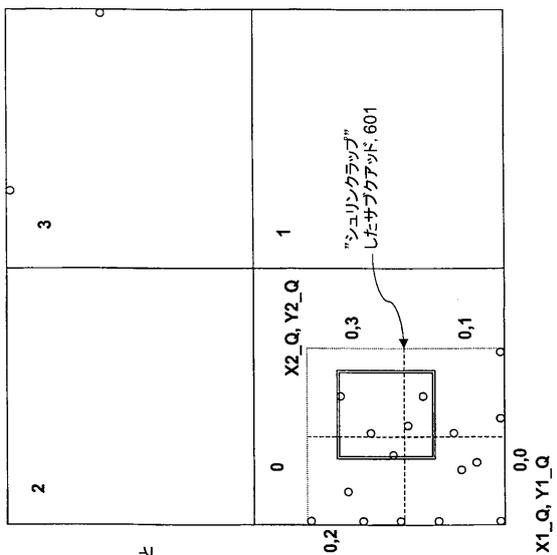


【 図 16 】



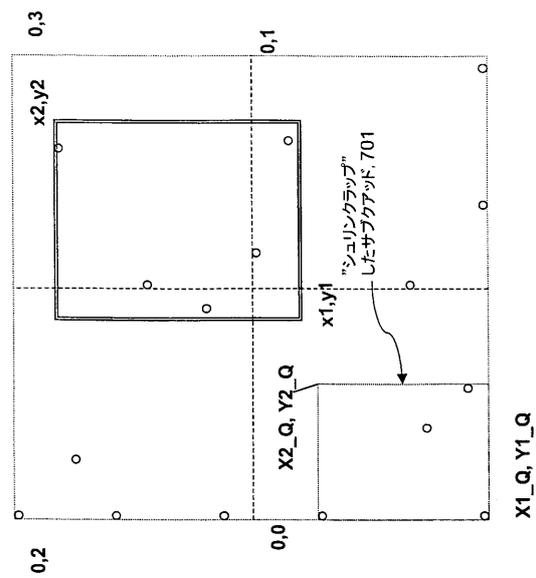
S13.1 一番外側のクアッドを検索
 S13.2 一番外側のクアッドと比較して関心のある区域の中心を調べる。条件S13.2が満足されなければクロープ。
 S13.3へ進む
 S13.3 条件満足せず
 S13.4 条件満足せず
 S13.5 次のレベルでクアッドを検索—クアッド0(図17参照)
 関心のある区域

【 図 17 】



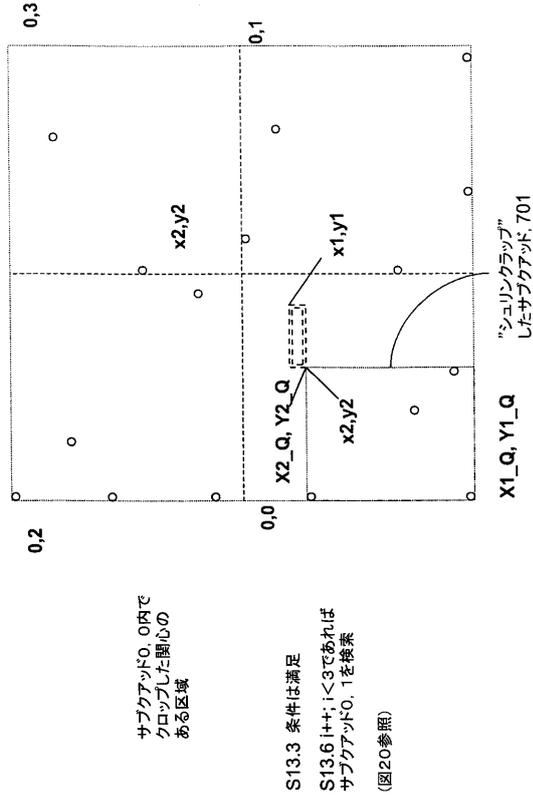
S13.1 データベースからサブクアッド0を検索
 S13.2 一番外側のクアッドと比較して関心のある区域の中心を調べる。条件S13.2を満足しなければクロープなし。
 S13.3へ進む
 S13.3 条件満足せず
 S13.4 条件満足せず
 S13.5 次のレベルでサブクアッドを検索—クアッド0, 0(図18参照)

【 図 18 】

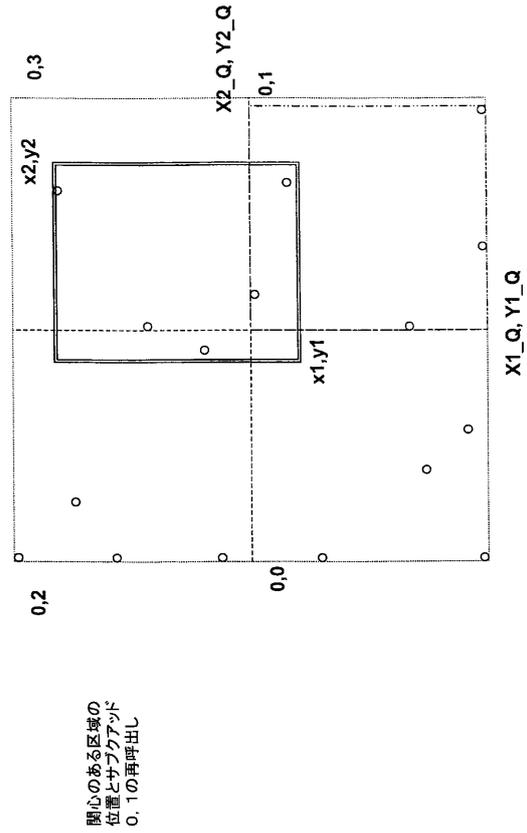


サブクアッド0の拡大図
 S13.1 データベースからサブクアッドを検索
 S13.2 関心のある区域の中心を調べる。
 $x2 > X2_Q \ \& \ y2 > Y2_Q$
 関心のある区域をクロープ
 $x2 = X2_Q \ \& \ y2 = Y2_Q$
 S13.3へ進む
 (図19参照)

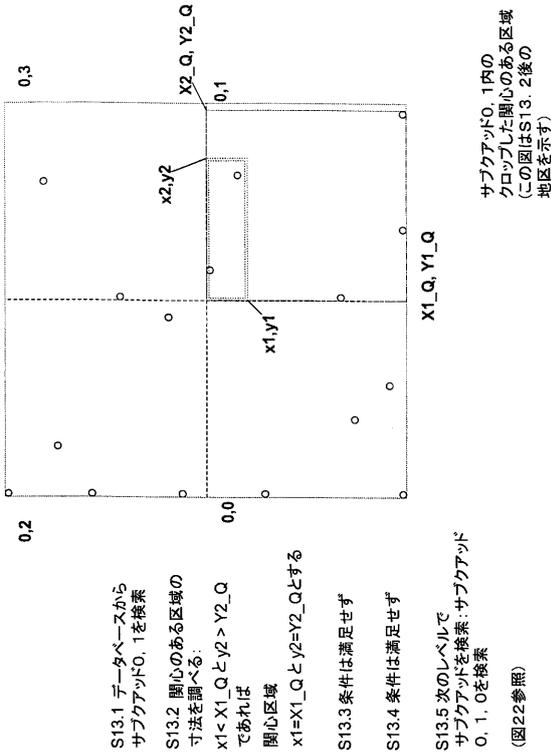
【 図 19 】



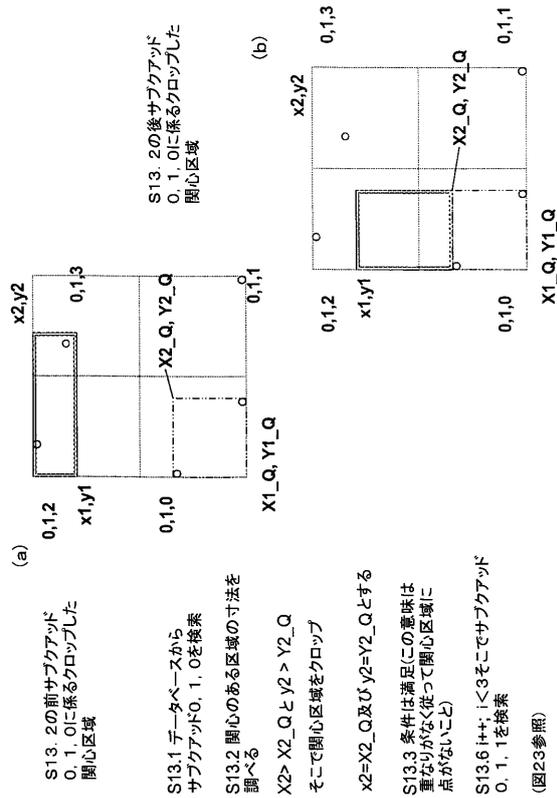
【 図 20 】



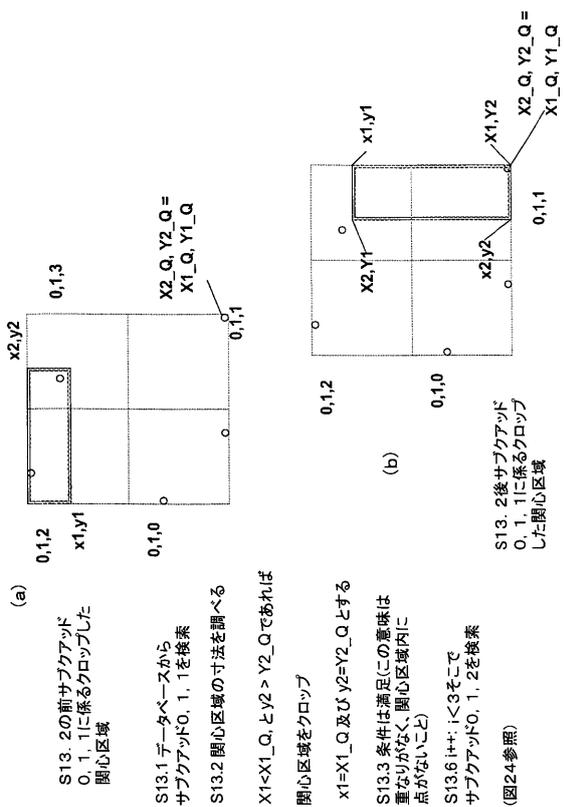
【 図 21 】



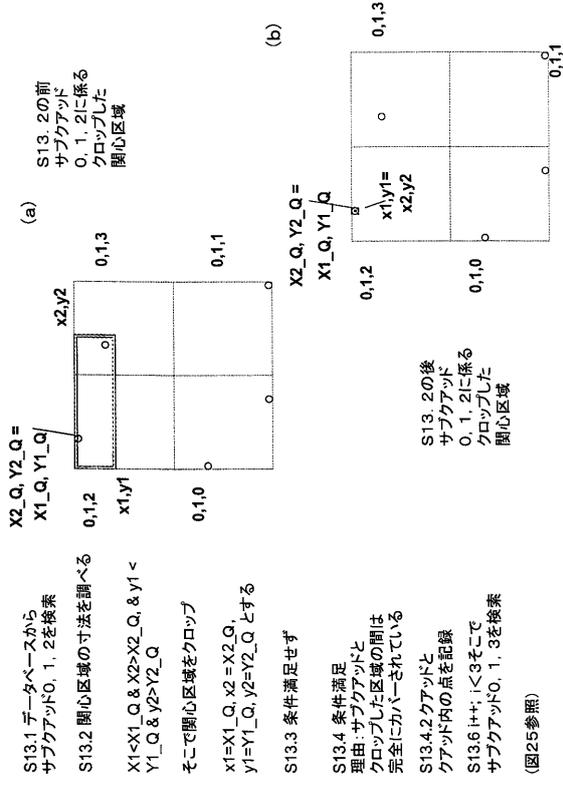
【 図 22 】



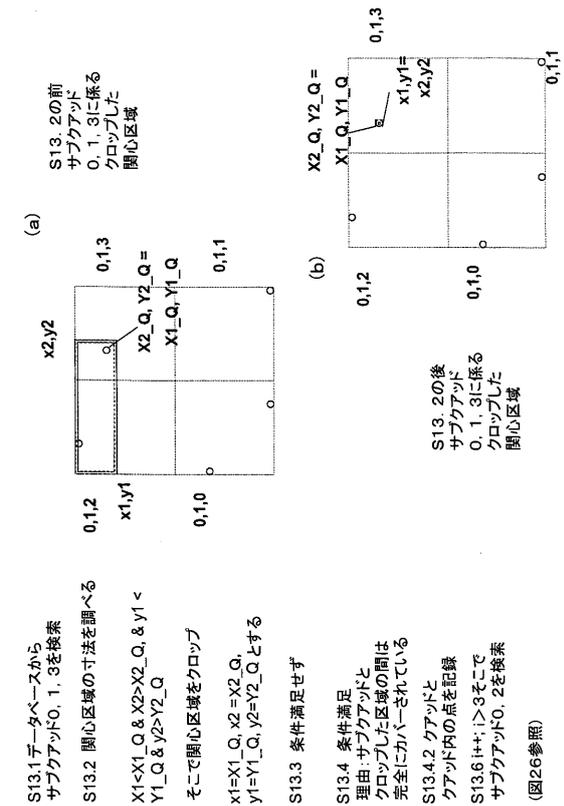
【 図 2 3 】



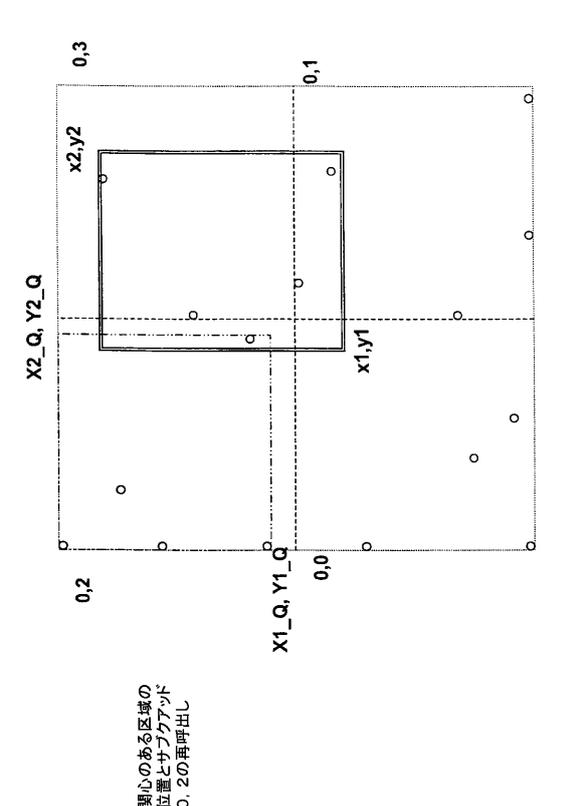
【 図 2 4 】



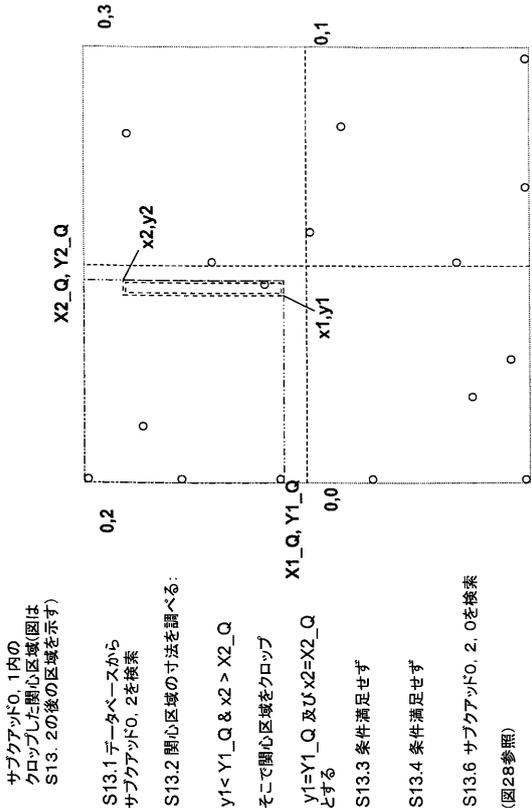
【 図 2 5 】



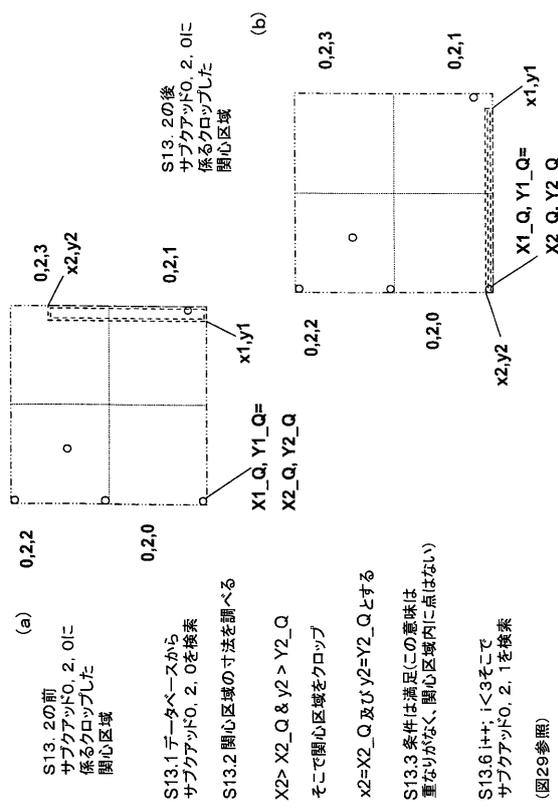
【 図 2 6 】



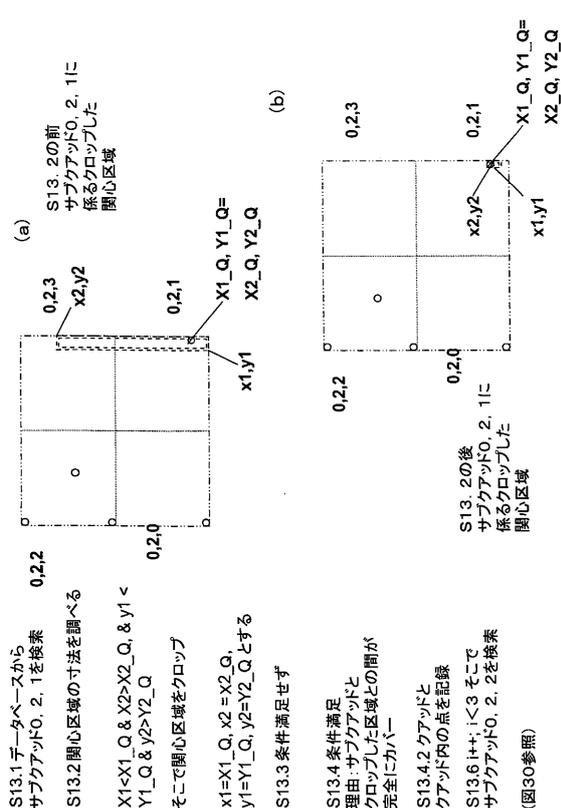
【 図 27 】



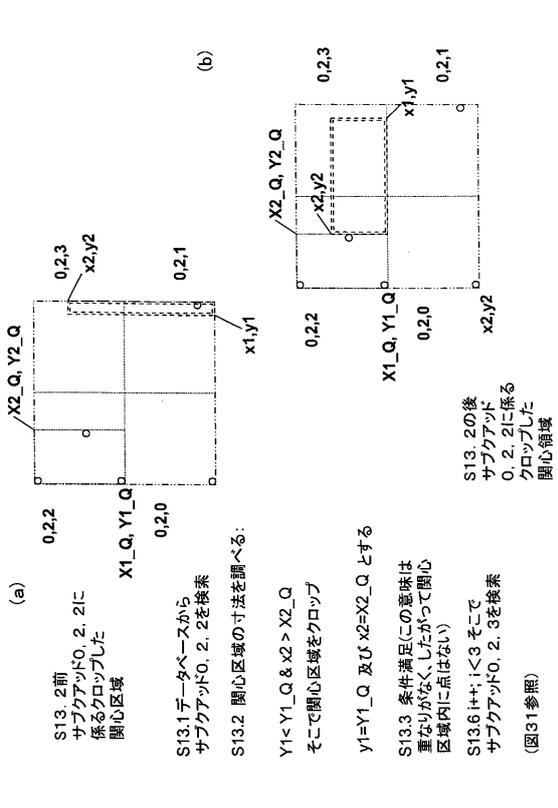
【 図 28 】



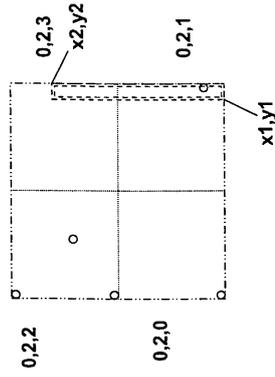
【 図 29 】



【 図 30 】



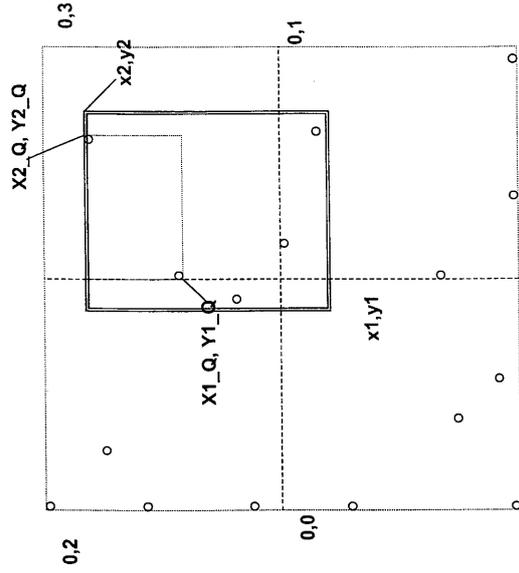
【図 3 1】



サブクアド
0, 2, 3に係る解析

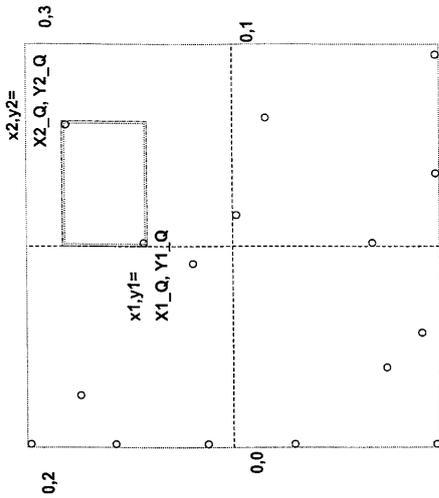
S13.1 データベースから
サブクアド0, 2, 3を検索:
このレベルではクアドはない、
そこで
S13.6.2: クアド0, 3を検索

【図 3 2】



関心のある位置と
サブクアド
0, 3の再呼出し

【図 3 3】



サブクアド0, 3における
クロップした関心区域(図は
S13. 2の後の区域を示す)

S13.1 データベースから
サブクアド0, 3を検索

S13.2 関心区域の寸法を調べる

$x1 < x1_Q \& y1 < y1_Q \& x2 >$
 $x2_Q \& y2 > y2_Q$

そこで関心区域をクロップ

$x1 = x1_Q, x2 = x2_Q,$
 $y1 = y1_Q, y2 = y2_Q$ とする

S13.3 条件満足せず

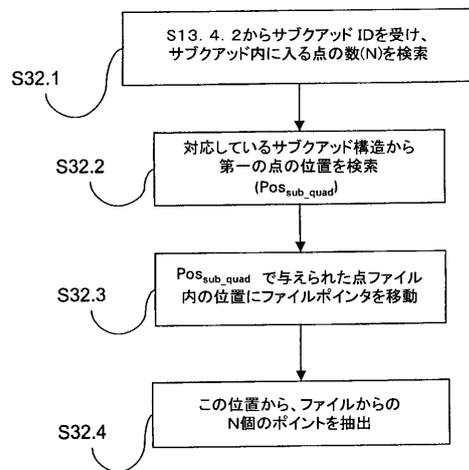
S13.4 条件満足

S13.4.2 クアドと
クアド内の点を記録

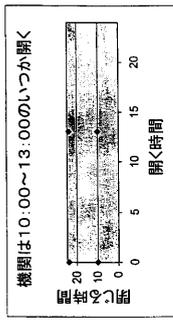
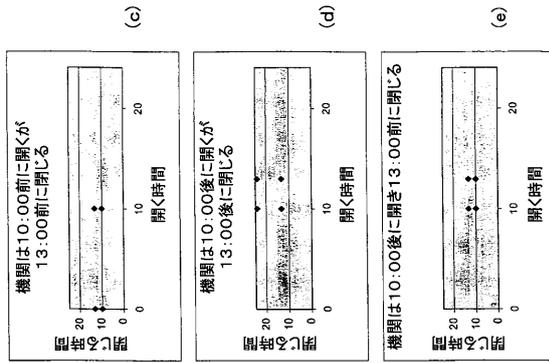
S13.6 i++; i > 3 まで
サブクアド1を検索

サブクアド1(及び2と3)についてはサブクアド0についての
上記段階を繰返すただし関心区域は他のいずれの
クアド内にも含まれないことに注意

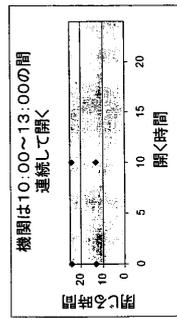
【図 3 4】



【 図 3 5 】



(a)



(b)

フロントページの続き

(31)優先権主張番号 0123154.7

(32)優先日 平成13年9月26日(2001.9.26)

(33)優先権主張国 英国(GB)

(74)代理人 100109830

弁理士 福原 淑弘

(74)代理人 100084618

弁理士 村松 貞男

(74)代理人 100092196

弁理士 橋本 良郎

(72)発明者 クラブツリー、イアン・バリー

イギリス国、アイピー9・2エヌエヌ、サフォーク、イプスウィッチ、タッティングストーン・ホワイト・ホース、ジ・オールド・コテージス 3

審査官 池田 聡史

(56)参考文献 特開2000-035965(JP,A)

吉田忠城 他, データ圧縮型多次元空間インデックスVA-TREE, 情報処理学会論文誌, 日本, 社団法人情報処理学会, 2000年10月15日, 第41巻 第SIG6(TOD7)号, pp.1~11

柳沼良知 他, 外接長方形上の点を用いた図形の間表現の一提案, 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, pp.821~829

丸山幹夫 他, 時変オブジェクトの管理を考慮した時空間データの索引手法, データベースとWeb情報システムに関するIPSJ DBS/ACM SIGMOD Japan Chapter/JSPS-RFTFAMCP 合同シンポジウム論文集, 日本, 社団法人情報処理学会, 2000年12月 6日, 第2000巻 第14号, pp.251~258

櫻井保志 他, 多次元空間における類似探索手法の提案, 情報処理学会研究報告, 日本, 社団法人情報処理学会, 1999年 7月23日, 第99巻 第61号, pp.103~108

(58)調査した分野(Int.Cl., DB名)

G06F 17/30