

(12) 特許協力条約に基づいて公開された国際出願

(19) 世界知的所有権機関
国際事務局

(43) 国際公開日
2016年5月12日(12.05.2016)

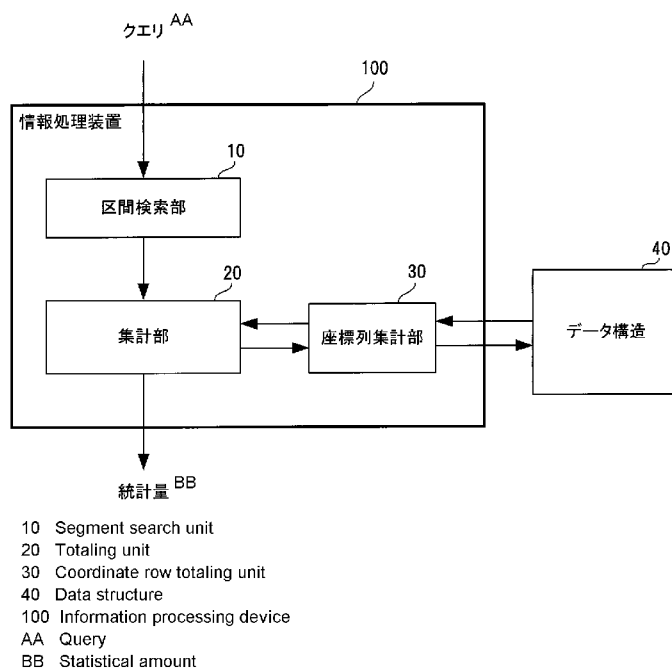


(10) 国際公開番号
WO 2016/072249 A1

- (51) 国際特許分類:
G06F 17/30 (2006.01)
- (21) 国際出願番号: PCT/JP2015/079476
- (22) 国際出願日: 2015年10月19日(19.10.2015)
- (25) 国際出願の言語: 日本語
- (26) 国際公開の言語: 日本語
- (30) 優先権データ:
特願 2014-227041 2014年11月7日(07.11.2014) JP
- (71) 出願人: NECソリューションイノベータ株式会社(NEC SOLUTION INNOVATORS, LTD.) [JP/JP]; 〒1368627 東京都江東区新木場一丁目18番7号 Tokyo (JP).
- (72) 発明者: 岡嶋 穰(OKAJIMA, Yuzuru); 〒1368627 東京都江東区新木場一丁目18番7号 NECソリューションイノベータ株式会社内 Tokyo (JP). 丸山 晃一(MARUYAMA, Kouichi); 〒1368627 東京都江東区新木場一丁目18番7号 NECソリューションイノベータ株式会社内 Tokyo (JP).
- (74) 代理人: 特許業務法人ブライトス(BRIGHTAS IP ATTORNEYS); 〒5300057 大阪府大阪市北区曽根崎2丁目5番10号 Osaka (JP).
- (81) 指定国 (表示のない限り、全ての種類の国内保護が可能): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.
- (84) 指定国 (表示のない限り、全ての種類の広域保護が可能): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), ユーラシア (AM, AZ, BY, KG, KZ, RU, TJ, TM), ユーロッパ (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).
- 添付公開書類:
— 国際調査報告 (条約第21条(3))

(54) Title: INFORMATION PROCESSING DEVICE, INFORMATION PROCESSING METHOD, AND COMPUTER-READABLE RECORDING MEDIUM

(54) 発明の名称: 情報処理装置、情報処理方法、及びコンピュータ読み取り可能な記録媒体



(57) Abstract: An information processing device 100 is provided with: a segment search unit 10 that, when a specific multidimensional area is specified as a query area in a data structure representing a set of points in a multidimensional space, specifies a segment configured only from points in a row of points obtained from the set of points, the coordinates of said row of points in each dimension being included in the query area except in one excluded dimension; a totaling unit 20 that specifies the range of the coordinate values in the one excluded dimension as a condition for including points that appear within the segment in the query area; and a coordinate row totaling unit 30 that calculates a statistical amount for a coordinate row obtained by using the specified segment and the range of the coordinate values as input and removing the coordinates for each point in the excluded one dimension, said statistical amount relating to a set of points in which all the coordinates correspond to all the coordinates that appear in the input segment and that are included in the input range.

(57) 要約:

[続葉有]



情報処理装置 100 は、多次元空間上の点の集合を表現するデータ構造を対象とし、クエリ領域として特定の多次元の領域が指定された場合に、点の集合から得られた点の列上にあり、且つ 1 つの次元を除いた残りの各次元の座標がクエリ領域に含まれている点のみで構成された区間を特定する区間検索部 10 と、区間内に出現する点がクエリ領域に含まれるための条件として、除かれた 1 つの次元での座標値の範囲を特定する集計部 20 と、特定された区間及び座標値の範囲を入力として、各点の除かれた 1 つの次元での座標を取り出して得られる座標列に関して、入力区間に出現し且つ値が入力範囲に含まれる全座標について、全座標が対応する点の集合に関する統計量を計算する座標列集計部 30 とを備えている。

明 細 書

発明の名称：

情報処理装置、情報処理方法、及びコンピュータ読み取り可能な記録媒体 技術分野

[0001] 本発明は、情報処理装置、情報処理方法、及びこれらを実現するためのプログラムを記録したコンピュータ読み取り可能な記録媒体に関し、特に、多次元データ上での効率的な検索を行うための、情報処理装置、情報処理方法、及びコンピュータ読み取り可能な記録媒体に関する。

[0002] 多次元空間上に点が大量に存在するときに、指定した矩形の範囲に包含される点を見つけることは、矩形範囲検索 (orthogonal range search) と呼ばれている。たとえば、次元数を d とおくと、 d 次元の多次元空間上の点は、 d 個の座標の組み合わせにより $p=(p_1, p_2, \dots, p_d)$ という形で表現できる。このような多次元空間上の点の集合が予め与えられているとする。さらに、個々の点 p が重み $w(p)$ を付与されているとする。

[0003] このとき、各次元 k の範囲を $[l_{qk}, u_{qk}]$ で表すとして、 $Q=[l_{q1}, u_{q1}] \times [l_{q2}, u_{q2}] \times \dots \times [l_{qd}, u_{qd}]$ という d 次元の矩形の範囲を考える。この矩形の範囲をクエリ領域と呼ぶこととすると、矩形範囲検索の目標は、このクエリ領域 Q に包含される点 p 、すなわち $\forall k \in \{1, \dots, d\}: l_{qk} \leq p_k \leq u_{qk}$ を満たす点 p の集合を求め、その集合に関する情報を計算することである。このとき、点 p がクエリ領域 Q に含まれるための d 個の条件 $\forall k \in \{1, \dots, d\}: l_{qk} \leq p_k \leq u_{qk}$ をクエリの範囲条件と呼ぶこととする。

[0004] このような矩形範囲検索は、地理情報を扱うアプリケーション、更には多次元データ分析において、重要な役目を果たす。以下に具体例を示す。

[0005] たとえば、地図上でのレストランの位置は（緯度，経度）という2つの値の組み合わせである2次元データで表すことができる。このとき、矩形範囲検索を用いると、経度が138度から139度、緯度が35度から36度の範囲に収まるような全てのレストランを検索できる。

- [0006] また、たとえば、ある会社の社員に関する統計データを（年齢、身長、年収）という3次元データで表すことができる。このとき、矩形範囲検索を用いると、年齢が30歳から40歳で、身長が170cmから180cm、年収が500万円から600万円の範囲に収まる全ての社員を検索できる。
- [0007] 更に、矩形範囲検索には、検索結果として何を返すかによって様々なバリエーションが存在する。バリエーションの一例として、レポート・クエリ（report query）と集計クエリ（aggregate query）とが挙げられる。
- [0008] まず、レポート・クエリは、クエリ領域に包含される全ての点のリストを返す矩形範囲検索である。レポート・クエリは、クエリ領域に包含される点の数をヒット数と呼ぶことにすると、ヒット数に比例する大きさのリストを返してしまうため、ヒット数が大きくなるような大規模データの分析には向かない。たとえば、数千万個の点が包含されるとき、数千万個の点を全て出力することになってしまう。
- [0009] そこで、大規模データの分析においては、クエリ領域に包含される全ての点のリストを返すよりも、それらの点について集計した結果を返す集計クエリが重要となる。様々な集計クエリの中でも最も代表的なクエリはカウント・クエリ（count query）である。
- [0010] このカウント・クエリは、クエリ領域に包含される点の数を返す矩形範囲検索である。この他にも、点にそれぞれ重みが付与されている場合に、クエリ領域に包含される点の重みの合計を返す合計（sum）クエリ、及び最大値を返す最大値（max）クエリなどが存在する。
- [0011] 本明細書では、これらのクエリが返す情報を、統計量と総称する。たとえば、統計量としては、カウント及び合計などが挙げられる。さらに、クエリに含まれる点の部分集合に関する統計量を「部分的な統計量」と呼び、クエリに含まれる全ての点の集合に関する統計量を「全体的な統計量」と呼ぶ。
- [0012] さて、矩形範囲検索に使うことができる代表的なデータ構造として、kd木が知られている（例えば、非特許文献1参照。）。kd木のサイズは、 $O(n)$ 、つまり線形サイズで表現できる。また、kd木における矩形範囲検索の最悪時

間計算量は、 $O(n^{(d-1)/d})$ であることが知られている。なお、 n はデータ数、 d は次元数である。kd木が達成している最悪時間計算量 $O(n^{(d-1)/d})$ は、これまで知られている実用的な線形サイズのデータ構造の時間計算量の中では、最良のものである。

[0013] また、サイズが $O(n)$ を超えてしまうような超線形サイズのデータ構造に、矩形範囲検索を適用した場合は、計算時間（計算量）を改善することができる。このような超線形サイズのデータ構造としては、たとえば、range tree と呼ばれるデータ構造が挙げられる。

[0014] さらに、矩形範囲検索は、ウェーブレット木 (wavelet tree) という2次元のデータ構造でも実現できる（例えば、非特許文献1参照）。この場合、2次元空間において探索が行なわれ、時間計算量は、 $O(\log n)$ 時間となる。

[0015] なお、上述したkd木およびウェーブレット木を用いた矩形範囲検索については、非特許文献1に詳しく記載されている。また、ウェーブレット木を用いて2次元空間上の統計量を計算する手法については、非特許文献2に詳しく記載されている。

先行技術文献

非特許文献

- [0016] 非特許文献1: Meng He, " Succinct and Implicit Data Structures for Computational Geometry" , Lecture Notes in Computer Science Volume 8066 "Space-Efficient Data Structures, Streams, and Algorithms" , pp 216-235, 2013, Springer Berlin Heidelberg, ISBN 978-3-642-40272-2
- 非特許文献2: Gonzalo Navarro and Luis M. S. Russo. "Space-efficient data-analysis queries on grids" , In Proceedings of the 22nd International Conference on Algorithms and Computation, ISAAC'11, pp. 323-332 , Berlin, Heidelberg, 2011. Springer-Verlag.

発明の概要

発明が解決しようとする課題

[0017] このように、矩形範囲検索は種々のデータ構造で実現することができるが、実際には、以下に示す問題がある。まず、矩形範囲検索をkd木で実現した場合においては、達成される最悪時間計算量 $O(n^{(d-1)/d})$ が、データ数 n 及び次元数 d のいずれか又は両方が大きくなる程、大きくなってしまいう問題がある。

[0018] また、矩形範囲検索を超線形サイズのデータ構造で実現した場合は、kd木で実現した場合に比べ、計算時間の改善は図られるが、超線形サイズのデータ構造のサイズが大きすぎるため、実際のアプリケーションに用いることが難しく、実用的でないという問題がある。

[0019] 更に、矩形範囲検索をウェーブレット木で実現した場合は、ウェーブレット木が2次元のデータにしか用いることができないため、3次元以上の任意の次元のデータ構造に対する検索ができないという問題がある。

[0020] 本発明の目的の一例は、上記問題を解消し、任意の次元について、線形サイズで、kd木よりも高速な矩形範囲検索を実現し得る、情報処理装置、情報処理方法、及びコンピュータ読み取り可能な記録媒体を提供することにある。

課題を解決するための手段

[0021] 上記目的を達成するため、本発明の一側面における情報処理装置は、多次元空間上の点の集合を表現するデータ構造を処理対象とする情報処理装置であって、

クエリ領域として、特定の多次元の領域が指定された場合に、前記点の集合を一行に並べて得られた点の列上にあり、且つ、前記多次元空間を構成する全次元のうち1つの次元を除いた残りの各次元の座標がクエリ領域に含まれている点のみによって構成されている、区間を特定する、区間検索部と、

前記区間検索部で特定された区間について、当該区間に出現する点が前記クエリ領域に含まれるための条件として、除かれた前記1つの次元における座標の値の範囲を特定する、集計部と、

前記区間検索部で特定された区間、及び前記集計部で特定された前記座標の値の範囲を入力として、

前記点の列の並び順と同じ順序で、前記点の集合の各点における、除かれた前記1つの次元での座標を取り出すことで得られる、座標列に関して、当該座標列において入力された前記区間に出現し、且つ、値が入力された前記範囲に含まれる、全ての座標について、

前記全ての座標が対応する点の集合に関する統計量を計算する、座標列集計部と、

を備えている、ことを特徴とする。

[0022] また、上記目的を達成するため、本発明の一側面における情報処理方法は、多次元空間上の点の集合を表現するデータ構造を処理対象とする情報処理方法であって、

(a) クエリ領域として、特定の多次元の領域が指定された場合に、前記点の集合を一行に並べて得られた点の列上にあり、且つ、前記多次元空間を構成する全次元のうち1つの次元を除いた残りの各次元の座標がクエリ領域に含まれている点のみによって構成されている、区間を特定する、ステップと、

(b) 前記(a)のステップで特定された区間について、当該区間に出現する点が前記クエリ領域に含まれるための条件として、除かれた前記1つの次元における座標の値の範囲を特定する、ステップと、

(c) 前記(a)のステップで特定された区間、及び前記(b)のステップで特定された前記座標の値の範囲を入力として、

前記点の列の並び順と同じ順序で、前記点の集合の各点における、除かれた前記1つの次元での座標を取り出すことで得られる、座標列に関して、当該座標列において入力された前記区間に出現し、且つ、値が入力された前記範囲に含まれる、全ての座標について、

前記全ての座標が対応する点の集合に関する統計量を計算する、ステップと、

を有する、を特徴とする。

[0023] 更に、上記目的を達成するため、本発明の一側面におけるコンピュータ読み取り可能な記録媒体は、多次元空間上の点の集合を表現するデータ構造を処理対象とする情報処理をコンピュータによって行なうためのプログラムを記録したコンピュータ読み取り可能な記録媒体であって、
前記コンピュータに、

(a) クエリ領域として、特定の多次元の領域が指定された場合に、
前記点の集合を一行に並べて得られた点の列上にあり、且つ、前記多次元空間を構成する全次元のうち1つの次元を除いた残りの各次元の座標がクエリ領域に含まれている点のみによって構成されている、区間を特定する、ステップと、

(b) 前記(a)のステップで特定された区間について、当該区間に出現する点が前記クエリ領域に含まれるための条件として、除かれた前記1つの次元における座標の値の範囲を特定する、ステップと、

(c) 前記(a)のステップで特定された区間、及び前記(b)のステップで特定された前記座標の値の範囲を入力として、

前記点の列の並び順と同じ順序で、前記点の集合の各点における、除かれた前記1つの次元での座標を取り出すことで得られる、座標列に関して、当該座標列において入力された前記区間に出現し、且つ、値が入力された前記範囲に含まれる、全ての座標について、

前記全ての座標が対応する点の集合に関する統計量を計算する、ステップと、

を実行させる命令を含む、プログラムを記録していることを特徴とする。

発明の効果

[0024] 以上のように、本発明によれば、任意の次元について、線形サイズで、kd木よりも高速な矩形範囲検索を実現することができる。

図面の簡単な説明

[0025] [図1]図1は、本発明の実施の形態における情報処理装置の概略構成を示すブ

ブロック図である。

[図2]図2は、本発明の実施の形態における情報処理装置の具体構成を示すブロック図である。

[図3]図3は、kd木の元になる2次元平面の一例を示している。

[図4]図4(a)は、2次元空間から得られるkd木の一例を示し、図4(b)は、kd木から得られる点の列Pの一例を示している。

[図5]図5は、本発明の実施の形態で用いられるウェーブレット木の一例を示す図であり、図5(a)及び(b)はそれぞれ次元が異なるウェーブレット木を示している。

[図6]図6は、本発明の実施の形態における情報処理装置の動作を示すフロー図である。

[図7]図7は、再帰的に区間を検索する関数 $\text{find_intervals}(v, Q)$ の動作を示すフロー図である。

[図8]図8は、座標列に基づき集計する関数 $\text{aggregate_interval}(v, s, e, l_{qf}, u_{qf})$ の動作を示すフロー図である。

[図9]図9は、2次元の場合の探索ノード数と包含次元数との変化を示す図である。

[図10]図10は、本発明と従来手法との計算量の比較を示す図である。

[図11]図11は、本発明の実施の形態における情報処理装置を実現するコンピュータの一例を示すブロック図である。

発明を実施するための形態

[0026] (発明の原理)

最初に、本発明の基本的な原理について、一般的なkd木を例に挙げて以下に説明する。

[0027] まず、kd木は、多次元データを扱うための二分探索木である。kd木の特徴は、空間全体を、次元1から次元dまでの各次元で順番に2分割していくことである。kd木では、木構造が空間の再帰的な分割を表しており、二分探索木の各ノードが部分領域に関連付けられている。各ノードvが関連付けられてい

る部分領域 $R(v)$ を、本明細書では、ノードの「カバー領域」と呼ぶことにする。カバー領域 $R(v)$ は、 $R(v)=[l_{v1}, u_{v1}] \times [l_{v2}, u_{v2}] \times \cdots \times [l_{vd}, u_{vd}]$ という d 次元の矩形の範囲として表すことができる。kd木においては、ノード v をルートとする部分木に存在する点は、 v のカバー領域 $R(v)$ に含まれる。

[0028] さらに、kd木は、各ノードに、そのノードをルートとする部分木に含まれる点の集合の統計量を保持しておくことができる。たとえば、カウント・クエリを高速に計算したい場合は、各ノードをルートとする部分木に含まれる点の数を、そのノードに保存する。

[0029] kd木における矩形範囲検索は以下のように実現される。まず、木全体のルートノードを出発点として、各内部ノードにおいて子ノードが関連付けられているカバー領域がクエリ領域と重なるかどうか判定され、重なる場合にだけ、その子ノードに移動することが繰り返される。子ノードへの移動は、特定の次元でカバー領域を二分割することに相当する。そして、ノードが関連付けられているカバー領域が、クエリ領域に完全に包含された場合は、そのノードに保存されている、部分木に含まれる点の統計量を、部分的な統計量として記憶する。何故なら、部分木に含まれる点はカバー領域に含まれ、当該点は、このときクエリ領域にも含まれるからである。この統計量は、クエリ領域に含まれる点の部分集合に関する統計量であるため、部分的な統計量である。

[0030] カバー領域がクエリ領域に完全に包含されるノードの統計量を全て見つけたとき、ノードの探索は終了する。このとき、これらの部分的な統計量を全て集計することにより、クエリ領域に含まれる全ての点に関する全体的な統計量を計算して出力する。

[0031] より厳密に説明するため、以下のように用語を定義する。次元 k において、 $[l_{vk}, u_{vk}] \subseteq [l_{qk}, u_{qk}]$ という範囲条件が成り立つとき、「カバー領域 $R(v)$ がクエリ領域 Q に次元 k で包含されている」と呼ぶ。次元 k においてカバー領域 $R(v)$ がこの範囲条件 $[l_{vk}, u_{vk}] \subseteq [l_{qk}, u_{qk}]$ を満たすとき、カバー領域に含まれる点も同様に次元 k の範囲条件 $p_k \subseteq [l_{qk}, u_{qk}]$ を満たす。何故なら、 $p_k \subseteq [l_{vk},$

$u_{vk}] \subseteq [l_{qk}, u_{qk}]$ が成り立つからである。すなわち、カバー領域に関して次元 k の範囲条件が成り立つとき、カバー領域に含まれる点に関しても次元 k の範囲条件が成り立つ。

[0032] さらに、 d 個の次元のうち h 個の次元において包含されているとき、「カバー領域の包含次元数は h である」、と定義する。また、包含次元数が d であるとき、すなわち、全ての次元が包含されているとき、カバー領域は完全に包含されていると呼ぶ。包含次元数は、 d 個の範囲条件のうち、満たされている条件の数である。

[0033] kd 木は、包含次元数が d に達するまで、すなわち、この範囲条件が d 個全て満たされるまで空間を分割する手法である。

[0034] kd 木の探索では、以上のように、与えられたクエリ領域にカバー領域が完全に含まれるノードの統計量を合わせることによって、探索結果が得られている。このとき、 kd 木はカバー領域がクエリ領域に一部包含されているが、完全に包含されていないノードを全てたどる必要があるが、このようなノードの数は $O(n^{(d-1)/d})$ になることが知られている。このため、 kd 木の最悪時間計算量は $O(n^{(d-1)/d})$ になる。

[0035] これに対して、本発明は、カバー領域がクエリ領域に完全に包含される前に kd 木での探索をやめ、ウェーブレット木での探索に移行することを特徴とする。

[0036] より厳密に言えば、本発明では、包含次元数が d に達するまで空間を分割するのではなく、包含次元数が $d-1$ に達するまで空間を分割する。このとき、まだ範囲条件が満たされていない最後の次元 f ($1 \leq f \leq d$) について、次元 f に関するウェーブレット木を用いて、次元 f の範囲条件を満たす座標を見つけることで、高速な検索を実現する。

[0037] これにより、本発明によれば、 kd 木を最後までたどる従来の手法と異なり、たどるノードの数が削減され、 kd 木より高速な矩形範囲検索が実現される。

[0038] (本明細書で用いられる概念)

ここで、本明細書で用いられる種々の概念について以下に説明する。本明細書では、全ての点の座標 p_i が $[0, n-1]$ の整数で表されるものとする。さらに、これらの整数は、二進表現で長さ $l = \text{ceil}(\log n)$ のビットで表されるとする。なお、 $\text{ceil}()$ は天井関数を表す。 $\log$ は底を2とする対数関数を表す。

[0039] たとえば、 $n=8$ のとき、全ての座標は $[0, 7]$ の整数で表され、二進表現では長さ $l = \text{ceil}(\log n) = 3$ ビットで表される。すなわち、 $0 = "000"$ 、 $1 = "001"$ 、 $2 = "010"$ 、 $3 = "011"$ 、 $4 = "100"$ 、 $5 = "101"$ 、 $6 = "110"$ 、 $7 = "111"$ で表すことができる。

[0040] ただし、本発明は、座標が整数で表されていない一般の多次元空間に対しても適用できる。たとえば、順位空間(rank space)への変換という手法を用いれば、任意の実数で表される n 個の点を、 $[0, n-1]$ の範囲の整数の座標に変換でき、その座標を用いることで、矩形範囲検索を実現できる。従って、この順位空間への変換を用いることにより、本発明を、実数で表される一般の多次元空間に適用することが可能である。なお、順位空間への変換については、たとえば上述の非特許文献1に記載されている。

[0041] また、本発明は、値を1と0の二進表現で表現できていれば、順位空間への変換が行なわれていなくても適用できる。すなわち、データ数が n であるとき、座標の値の範囲が $[0, n-1]$ の範囲から外れているデータであっても、本発明は適用できる。本明細書では、計算量の理論的な分析を行うために $[0, n-1]$ の範囲に限定して説明しているだけであり、実用上は $[0, n-1]$ の範囲に限らなくても問題なく本発明を適用できる。

[0042] また、本明細書では、「接頭辞」という概念を用いる。接頭辞とは、整数を二進表現で表したときに、その上位ビットだけを取り出したものである。本明細書では、整数の上位 h ビットの接頭辞を、 h 個の1と0、および $(l-h)$ 個の*の組み合わせで表記する。*はワイルドカードであり、1および0のどちらでもよいことを表している。ある整数が特定の接頭辞から始まることは、その整数が、特定の連続範囲に含まれることに対応する。

[0043] たとえば、整数が長さ $l=3$ のビット列であらわされているとする。このとき

長さ1の接頭辞"0**"は、"000", "001", "010", "011"という4つの値に対応する。すなわち、この接頭辞は、整数上の値の範囲である["000", "011"]= $[0, 3]$ に対応する。同様に、長さ2の接頭辞"01*"は、"010"と"011"という2つの値に対応しており、値の範囲で表すと["010", "011"]= $[2, 3]$ に対応する。長さ l の接頭辞は、ひとつの整数だけに対応する。

[0044] また、本明細書では、列(sequence)に関して、以下の表記が用いられる。たとえば、長さ n の列 A が存在するとき、 A の最初の要素を $A[0]$ 、 A の最後の要素を $A[n-1]$ とする。さらに、 A 上の添字 s の要素 $A[s]$ から添字 e の要素 $A[e]$ までの $(e-s+1)$ 個の要素で構成される列を $A[s, e]$ で表し、終端 $A[e]$ を含まない場合は $A[s, e)$ で表すものとする。また、 $A[s, e]$ に含まれる要素を、 A 上で区間 $I=[s, e]$ に含まれる要素、と呼ぶ。

[0045] さらに、本明細書では、座標の値の範囲(range)と、列上の添字の区間(interval)とが、厳密に区別される。座標の値の範囲も、列上の添字の区間も、2つの数字のペアで表されるが、この明細書では、 $[l, u]$ を「範囲」と呼ぶとき、 l と u は座標の値である。一方、 $[s, e]$ を「区間」と呼ぶとき、 s と e は列に関する添字である。

[0046] (実施の形態)

続いて、本発明の実施の形態における、情報処理装置、情報処理方法、及びプログラムについて、図1～図10を参照しながら説明する。

[0047] [装置構成]

最初に、本実施の形態における情報処理装置の概略構成について図1を用いて説明する。図1は、本発明の実施の形態における情報処理装置の概略構成を示すブロック図である。図1に示す本実施の形態における情報処理装置100は、多次元空間上の点の集合を表現するデータ構造40を処理対象とする装置である。そして、図1に示すように、情報処理装置100は、区間検索部10と、集計部20と、座標列集計部30とを備えている。

[0048] このうち、区間検索部10は、クエリ領域として、特定の多次元の領域が指定された場合に機能する。クエリ領域は、上述したように、例えば、各次

元に対応する d 個の範囲の組み合わせで表現される。

[0049] 区間検索部 10 は、点の集合を一行に並べて得られる点の列 P 上にあり、且つ、多次元空間を構成する全次元のうち 1 つの次元を除いた残りの各次元の座標がクエリ領域に含まれている点のみによって構成されている、区間を特定する。言い換えると、区間検索部 10 は、点がクエリ領域に含まれるための d 個の条件のうち $(d-1)$ 個の条件を満たす、点を含む、列 P 上の添字の区間を 0 個以上特定する。また、区間検索部 10 は、特定した区間を集計部 20 に出力する。

[0050] 集計部 20 は、区間検索部 10 で特定された区間について、クエリ領域に含まれるための条件として、除かれた 1 つの次元における座標の値の範囲を特定し、特定した座標の値の範囲と区間検索部 10 で特定された区間とを座標列集計部 30 に出力する。

[0051] 言い換えると、集計部 20 は、区間検索部 10 で特定された点の列 P の各区間に含まれる点が満たしていない最後の範囲条件が対応する次元 f について、これらの点がクエリ領域に含まれるために必要な範囲条件となる座標の値の範囲を特定する。そして、集計部 20 は、次元 f に対応する座標列集計部 30 に、区間検索部 10 が特定した列 P 上の添字の区間と、次元 f の座標の値に関する範囲条件となる座標の値の範囲とを与えて、問い合わせを行う。

[0052] 座標列集計部 30 は、区間検索部 10 で特定された区間（添字の区間）と、次元 f における座標の値の範囲とが入力されると、機能する。座標列集計部 30 は、入力となされると、点の列 P の並び順と同じ順序で、点の集合の各点における次元 f での座標を取り出すことで得られる、座標列に関して、入力された区間に出現し、且つ、値が入力された範囲に含まれる、全ての座標について、これらの座標が対応する点の集合に関する統計量を計算する。また、座標列集計部 30 は、計算した統計量を集計部 20 に出力する。

[0053] このように、情報処理装置 10 では、クエリ領域を表す d 個の条件のうち $(d-1)$ 個の条件を満たされるまでだけ、多次元空間が分割されるので、 kd 木を探索する場合に比べて、クエリ領域の分割に要する計算量が削減される。この

ため、情報処理装置 10 によれば、任意の次元 d について、線形サイズで、 kd 木よりも高速な矩形範囲検索を実現することができる。

[0054] 次に、図 2 を用いて、本実施の形態における情報処理装置 100 の構成について更に具体的に説明する。図 2 は、本発明の実施の形態における情報処理装置の具体構成を示すブロック図である。

[0055] 図 2 に示すように、本実施の形態では、情報処理装置 100 は、上述した区間検索部 10 と、集計部 20 と、座標列集計部 30 と、に加えて、記憶部 43 と、入力受付部 50 と、出力部 60 と、を備えている。

[0056] また、本実施の形態では、次元毎に設けられた d 個の座標列集計部 $30-1 \sim 30-d$ が備えられている。そして、座標列集計部 $30-1 \sim 30-d$ それぞれは、対応する次元と区間検索部 10 で特定された区間の次元とが一致する場合に、点の集合に関する統計量を計算する。なお、以降の説明において、座標列集計部を個別に特定しない場合は、「座標列集計部 30」と表記する。

[0057] また、入力受付部 50 は、外部からのクエリ領域の入力を受け付け、これを区間検索部 10 に出力する。記憶部 43 は、データ構造 40 を記憶している。本実施の形態では、データ構造 40 は、区間検索部 10 による区間の特定に用いられる区間検索用データ構造 41 と、座標列集計部 30 による統計量の計算に用いられる座標列集計用データ構造 42 とを有している。

[0058] また、区間検索部 10 は、入力受付部 50 からクエリ領域が出力されてくると、記憶部 43 に対して問い合わせ、区間検索用データ構造 41 を取得する。区間検索用データ構造 41 は、クエリ領域が指定されたときに、区間検索部 10 によって、点の列 P 上における、クエリ領域を表す d 個の条件のうち ($d-1$) 個の条件を満たす点を含む区間を特定するための、データ構造である。

[0059] 本実施の形態では、区間検索用データ構造 41 としては、ノードを有する木構造によって表現されたデータ構造が挙げられる。また、このデータ構造において、ノードは、多次元空間に設定された複数のカバー領域のいずれかと、点の列上において、該当するカバー領域に含まれる点が出願する区間と

、の両方に関連付けられている。具体的には、区間検索用データ構造41としては、kd木が挙げられる。なお、本実施の形態において、区間検索用データ構造41は、kd木に限定されず、木構造の各ノードが矩形の領域に関連付けられた任意のデータ構造であれば良い。他の具体例としては、kDB木、R木、bounding volume hierarchy (BVH)、と呼ばれるデータ構造も挙げられる。

[0060] 本実施の形態では、区間検索部10は、ノードのうち、関連付けられたカバー領域に存在する点における、1つの次元を除いた残りの各次元の座標がクエリ領域に含まれている、ノードを特定する。そして、区間検索部10は、特定した1又は2以上のノードが関連付けられている区間を特定する。

[0061] ここで、区間検索用データ構造41について、更に詳しく説明する。上述したように、本実施の形態では、区間検索用データ構造41としてkd木を用いることができる。kd木は、各ノードが多次元空間上に設定された矩形領域に関連付けられた二分木である。この矩形領域が、上述したカバー領域に相当する。kd木のルートノードのカバー領域は、グリッド上の全領域、 $[0, n-1] \times [0, n-1] \times [0, n-1] \times \dots \times [0, n-1]$ である。また、各ノードの深さは、いずれかひとつの次元に着目して空間を二分割すると、1つ下がり、分割する次元は1, 2, 3, ..., dの順番で繰り返して選ぶものとする。

[0062] kd木は、ルートノードを出発点として、以下のように再帰的に構築できる。まず、各内部ノードにおいて、その深さで分割に用いられる次元がkであるとき、そのカバー領域に含まれる全ての点について次元kの座標を調べ、中央値となる座標を選び、その座標を用いてカバー領域を二分割する。すなわち、その座標をtとおくと、カバー領域は、次元kの座標がtより小さくなる領域と、次元kの座標がtと同じか、tより大きくなる領域とに分割される。

[0063] そして、この内部ノードの2つの子ノードは、分割によって得られた2つの領域に対応する。この分割を左の子ノードと右の子ノードに再帰的に適用することで、kd木は構築される。また、内部ノードはそれぞれ、分割に用いられた座標を保持する。従って、この分割を繰り返して、カバー領域に含まれる点が1つになれば、これ以上分割せずに、このカバー領域に関連付けら

れた葉ノードを構築して保持する。終端となる葉ノードは、そのカバー領域に含まれる点自体を保持する。また、各点が重みを持つ場合、葉ノードは、この重みについても同様に保持する。

[0064] また、kd木の各ノード v のカバー領域 $R(v)$ に関して、ノード v は、カバー領域の値を直接保持していても良いし、kd木を探索するときに、通過したノードに保持されている分割に用いられた座標から、カバー領域の値を動的に計算しても良い。

[0065] なお、kd木の定義方法には複数のバリエーションが存在するが、本実施の形態においては、そのいずれの定義が用いられても良い。また、本実施の形態では、kd木の内部ノードが分割に用いた座標のみを保持する定義を用いて説明が行われているが、これに限定される趣旨ではない。本実施の形態においては、kd木の内部ノードが分割に用いた点自体を保持する定義が用いられていても良い。また、後述するように、葉ノードが1つの点を保持する定義に限定されず、葉ノードが複数の点を保持する定義が用いられていても良い。

[0066] また、上述した点の列 P は、ノードに関連付けられたカバー領域それぞれに存在する点が、ひとつながりで連続して出現するように、点の集合に含まれる点を一列に並べて得られている。

[0067] 具体的には、点の列 P は、構築されたkd木を用いて以下のように定義される。まず、点の列 P において、各点は、kd木を通りがけ順(in-order)に探索したときに見つかる順序に基づいて並べられているものとする。すなわち、kd木のルートノードを出発点として、まず左の部分木が探索され、次にルートノード自身を通り、右の部分木が探索される。この探索順序を再帰的に全ノードに適用するとき、kd木に含まれる全ての点に一回ずつアクセスすることになるが、その順番通りに点を並べて得られる列を P とする。

[0068] このとき、kd木の任意のノード v に関して、以下の条件を満たす区間 $I_v = [s, e]$ が存在する。その条件とは、点の列 P 上で区間 I_v に含まれる点の集合が、 v をルートノードとする部分木に含まれる点の集合と一致することである。各

ノード v は、このような区間 I_v を保持するものとする。このとき、 v の部分木に含まれる点の数 n_v は、 $n_v=e-s+1$ という式で計算できる。

[0069] さらに、この点の列 P に対応する、 k 個の座標列 P_k を考える。ただし、 P_k は、次元 k について、点の列 P の並び順と同じ順序で、各点の次元 k での座標を取り出すことで得られる座標列である。

[0070] ここで、図3及び図4を用いて、kd木の具体例について説明する。また、以下の説明では、次元数 d は2次元であるとする。図3は、kd木の元になる2次元平面の一例を示している。図4(a)は、2次元空間から得られるkd木の一例を示し、図4(b)は、kd木から得られる点の列 P の一例を示している。

[0071] 図3に示すように、2次元平面上には複数の点が存在している。具体的には、 $[0,7] \times [0,7]$ のグリッドで表される2次元平面に、 $n=8$ 個の点が存在している。それぞれの点には0から7の番号が与えられているが、これは後述するように、点の列 P における順序を表している。すなわち、0と書かれた点は、点の列 P における最初の点である $P[0]$ を表している。また、グリッドの上に引かれている太線は、kd木のノードによって生じる空間の分割を表しており、横の太線は次元1に関する分割を、縦の太線は次元2に関する分割を表している。

[0072] また、図4(a)に示すように、図3に示した各点はkd木に格納される。この木構造において、深さが偶数のノードは次元1に関する分割を表し、深さが奇数のノードは次元2に関する分割を表している。内部ノードの上部に示されている式は、どの座標を用いて空間を分割しているかを表している。さらに、ノードの下部には、そのノードに対応する点の列 P での区間 I_v が示されている。葉ノードでは、分割に用いる座標の代わりに、座標の組で表される点が保持されている。

[0073] また、図4(b)に示すように、点の列 P は、図3に示した点と図4(a)に示したkd木とに対応する。定義により、座標列 P_1 と P_2 も同じ図で表される。図4(b)の一行目は添字 i の値を表し、二行目は座標列 P_1 、三行目は座標

列 P_2 を表す。この図によれば、たとえば、 $P[0] = (P_1[0], P_2[0]) = (0, 4)$ である。

[0074] 続いて、図4 (a) に示したkd木の一例が、上述したkd木の定義を満たしていることについて説明する。たとえば、kd木のルートノード v は、与えられたグリッドにおける全ての領域をカバー領域 $R(v)$ として持つ。すなわち、カバー領域 $R(v)=[0, 7] \times [0, 7]$ である。さらに、全ての点はルートノードの子として含まれるため、区間 $I_v = [s, e]=[0, 7]$ である。ルートノードは深さ0であり、次元1で空間を分割している。次元1の座標のうち中央値となる $p_1=4$ に注目して、 $p_1 < 4$ となる領域と $4 \leq p_1$ となる領域とに分割している。従って、左の子ノードのカバー領域は $[0, 3] \times [0, 7]$ であり、右の子ノードのカバー領域は $[4, 7] \times [0, 7]$ である。以下、同様に分割されて構築されている。

[0075] また、点の列 P は、kd木を通りがけ順に巡回した順番に並んでいるため、全てのノードについて、点の列 P 上で区間 I_v に含まれる点は、そのノードをルートとする部分木に含まれている。たとえば、図4 (a) において木全体のルートノードの左の子ノードは、 $I_v=[0, 3]$ であるが、このことは、 $P[0]$ から $P[3]$ の4つの点が、このノードをルートとする部分木に含まれることを表す。

[0076] また、本実施の形態では、座標列集計部30は、まず、座標列集計用データ構造42を用いて、座標列から得られる複数の部分列のうち、入力された範囲に含まれる座標のみが出現する部分列を特定する。そして、座標列集計部30は、特定した部分列上の区間であって、座標列において入力された区間中出现する座標が出現している、区間を特定し、特定した部分列上の区間中出现する座標が対応する点の集合に関する統計量を計算する。なお、後述するように、部分列としては、座標のビット表現が同じ接頭辞で始まる座標を、座標同士の位置関係を保ったまま抽出することで得られるものが挙げられる。

[0077] また、本実施の形態では、座標列集計用データ構造42は、1から d の各次元 k に対応する座標列 P_k を表現するデータ構造である。ただし、 P_k は、次元 k について、前記点の列 P の並び順と同じ順序で、各点の次元 k での座標を取り出

すことで得られる座標列である。座標列集計用データ構造42は、座標列集計部30において、座標列 P_k 上の添字の区間と、座標の値の範囲とが入力されたときに、座標列上での位置が入力された区間に含まれ、且つ、値が入力された範囲に含まれる、全ての座標について、これらの座標が対応する点の集合に関する統計量を計算できるようにする、データ構造である。

[0078] 座標列集計用データ構造42としては、上述の部分列に関連付けられた複数のノードを有するデータ構造が挙げられる。この場合、各ノードは、部分列において出現する各座標のビット表現における、一つ以上の特定の桁のビットを取り出し、取り出したビットを部分列と同じ順序で並べることによって得られる、ビットの列を用いて表現できる。この場合、座標列集計部30は、各ノードを表現するビットの列を用いて、部分列上の区間を特定する。

[0079] 具体的には、本実施の形態では、座標列集計用データ構造42として、ウェーブレット木を用いることができる。この場合、区間検索用データ構造42は、1から d の各次元に対応する、 d 個のウェーブレット木で構築されている。この d 個のウェーブレット木の集合を $W=\{w_k\}$ とする。

[0080] ただし、本実施の形態において、座標列集計用データ構造42は、ウェーブレット木に限定されるものではない。座標列集計用データ構造42は、整数列上の添字の区間と整数の値の範囲とが条件として与えられたとき、整数列上で、その区間に含まれ、かつ、その範囲条件を満たす、点を検索できる、データ構造であれば良い。たとえば、その他の座標列集計用データ構造42としては、Chazelleのcompressed range tree、compressed range treeを外部記憶に拡張したCompressed Range B-tree (CRB-tree)等が挙げられる。

[0081] ここで、上述した図3及び図4に加え、図5を用いて、座標列集計用データ構造42の具体例、すなわち、 d 個のウェーブレット木の具体例について説明する。また、以下の説明では、次元数は2次元であるとする。図5は、本発明の実施の形態で用いられるウェーブレット木の一例を示す図であり、図5(a)及び(b)はそれぞれ次元が異なるウェーブレット木を示している。

[0082] 図5 (a) は座標列 P_1 及び座標列 P_1 に対応するウェーブレット木 w_1 を示し、図5 (b) は座標列 P_2 及び座標列 P_2 に対応するウェーブレット木 w_2 を示している。また、各図の左に示されたテーブルは座標列を表している。1行目は列の添字 i を示し、2行目は添字に対応する整数を示している。そして、3行目以降には、各整数のビット表現が示されている。

[0083] そして、次元 k の座標列 P_k に対応するウェーブレット木は、以下のように、二分木として定義される。なお、ウェーブレット木は、深さ l の二分木である。この木構造において、親から左の子に向かうエッジはビット0に対応し、親から右の子に向かうエッジはビット1に対応する。

[0084] まず、ウェーブレット木のルートノードは、深さ0にあり、長さ0ビットの座標接頭辞に対応しているものとする。さらにウェーブレット木の深さ h にあるノード v は、ルートノードからそのノードへのパスに出現する h 個のビットを連結して得られる h ビットの座標接頭辞 π に対応しているものとする。深さ l にあるノードは全て葉ノードである。葉ノードは、ルートからそのノードへのパスに出現する l 個のビットを連結して得られる l ビットで表される1つの整数に対応する。

[0085] さらに、深さ h において座標接頭辞 π に対応するノード v は、座標列 P_k の部分列 $P_k(\pi)$ に対応する。ただし、 $P_k(\pi)$ は、座標列 P_k から座標接頭辞 π で始まる全ての整数を、その並び順と同じ順序を保ったまま抜き出した部分列とする。本明細書では、元となる P_k を「座標列」、座標接頭辞 π に注目して抜き出した部分列 $P_k(\pi)$ を、「座標部分列(coordinate subsequence)」と呼び分けることとする。

[0086] 座標部分列 $P_k(\pi)$ の添字 i の要素である $P_k(\pi)[i]$ が、元の座標列 P_k の添字 j の要素である $P_k[j]$ に相当するとき、この $P_k(\pi)[i]$ は、元々は、点 $P[j]$ の次元 k の座標である。このとき、本明細書では、座標 $P_k(\pi)[i]$ は、点 $P[j]$ に属する、と呼ぶこととする。

[0087] さらに、このノード v は、 $P_k(\pi)$ の各要素の $h+1$ 番目のビットのみを抜き出して同じ順序で連結したビット列 B_v を記憶しているものとする。すなわち、整

数 $P_k(\pi)[i]$ の $h+1$ ビット目が0であるとき $B_v[i]=0$ となり、1であるとき $B_v[i]=1$ となるようなビット列である。

[0088] 具体的には、図5(a)及び(b)に示すように、本実施の形態では、次元1の座標列 P_1 に対して構築されたウェーブレット木 w_1 と、次元2の座標列 P_2 に対して構築されたウェーブレット木 w_2 とが用いられる。また、図5(a)及び(b)においては、各ノードについて対応する座標接頭辞 π 、座標部分列 $P_k(\pi)$ 、およびビット列 B_v が示されている。

[0089] また、図5(a)及び(b)に示すように、ウェーブレット木 w_1 は、座標列 $P_1=(0, 2, 1, 3, 4, 7, 5, 6)$ のウェーブレット木である。座標列 P_1 の各要素は、3ビットで表されている。そして、各ウェーブレット木のルートノードは、座標接頭辞 $\pi="***"$ に結びつけられている。よって、この座標接頭辞は3ビットで表せる全ての値、すなわち、 $["000", "111"]=[0, 7]$ までの範囲の全ての値に対応している。このため、ルートノードは、座標部分列 $P_1(\pi)$ の $0+1=1$ ビット目のビットをビット列 B_v として保持している。

[0090] 次に、ルートノードの左の子ノードは、接頭辞 $"0**"$ に対応しており、1ビット目が0で始まる3ビットの整数、すなわち $[0, 3]$ の範囲に対応し、さらに、座標列 P_1 から $[0, 3]$ の範囲に該当する値だけを取り出した座標部分列 $P_1(\pi)=(0, 2, 1, 3)$ にも対応する。よって、この左の子ノードは、その2ビット目のビットをビット列 B_v として保持している。なお、以降の子ノードも同様に考えることができる。

[0091] また、ウェーブレット木は、各内部ノード v について、ビット列 B_v の完備辞書を保持している。完備辞書は、長さ n のビット列 B に対して access 、 rank 、 select と呼ばれる3種類の操作をサポートするデータ構造である。この3種類の操作は以下のように定義される。

[0092] $\text{access}(B, i)$ は、 B 上の添字 i の要素 $B[i]$ を返す。

$\text{rank1}(B, i)$ は、 $B[0, i]$ の範囲に存在する1の数を返す。

$\text{rank0}(B, i)$ は、 $B[0, i]$ の範囲に存在する0の数を返す。

$\text{select1}(B, i)$ は、 B 上で $i+1$ 番目の1が出現する位置 j を返す。

$\text{select0}(B, i)$ は、 B 上で $i+1$ 番目の0が出現する位置 j を返す。

[0093] なお、完備辞書は、文献によっては、簡潔ビットベクトルあるいはrank / select dictionaryなどと呼ばれることがある。

[0094] また、図5 (a) 及び (b) の例では、説明のため、ウェーブレット木の各ノードにおいて、座標接頭辞 π と、座標部分列 $P_k(\pi)$ と、ビット列 B_v とが、示されているが、実際には、ウェーブレット木は B_v の完備辞書だけを保持しており、座標接頭辞 π および座標部分列 $P_k(\pi)$ を保持している必要はない。これは、座標接頭辞 π は、たどってきたエッジの情報から計算でき、座標部分列 $P_k(\pi)$ の各要素は、ビット列 B_v の完備辞書を用いることで計算できるためである。よって、記憶部43には、実際には、座標列集計用データ構造42として、完備辞書のみが保持される。

[0095] なお、ウェーブレット木の定義方法は文献によって異なる。上述した非特許文献1では、ウェーブレット木は、接頭辞を用いずに定義されているが、本明細書では、ウェーブレット木は、説明のため、接頭辞を用いて定義されている。どちらの定義の場合であっても、ウェーブレット木の本質的な構造は同一であり、同じ動作を実現できる。

[0096] また、ウェーブレット木は、木構造としての探索が可能な構造、即ち、複数のノードを有する構造を有していれば、明示的に木構造として構成されていなくてもよい。たとえば、ウェーブレット行列という、ビット列をノードごとに分けずにウェーブレット木を実装する手法が知られているが、本発明における議論は、ウェーブレット行列を使った場合でもまったく同様に成立する。

[0097] また、区間検索部10によって特定された区間が複数である場合は、集計部20は、座標列集計部30によって計算された区間毎の統計量（即ち、部分的な統計量）を、更に集計する。この場合、集計部20は、集計によって得られた全体的な統計量を、クエリ領域に含まれる点の集合に関する全体的な統計量として、出力部60に出力する。その後、出力部60は、集計部20が出力した全体的な統計量を、外部の端末装置、サーバ装置等へ出力する。

。

[0098] [探索アルゴリズムの概要]

続いて、情報処理装置 100 の動作を説明する前に、情報処理装置 100 において用いられる探索アルゴリズムの概要について以下に説明する。

[0099] 初めに、kd木を用いて、ノードのカバー領域がクエリ領域と重なっており、かつ、ノードのカバー領域の包含次元数が $(d-1)$ になるノードを見つける。このことはすなわち、ノードのカバー領域がクエリ領域に含まれるための d 個の範囲条件のうち、 $(d-1)$ 個の範囲条件が成り立っていることを表す。ここで、条件が成り立っていない次元を f とおく。

[0100] このノード v が保持する添字の区間 $I_v = [s, e]$ に注目すると、点の列 P のうち $P[s, e]$ に含まれる点は、このノード v をルートとする部分木に含まれる点であるから、 f 以外の次元に関して、クエリの範囲条件を満たすことが保証される。ただし、次元 f に関する範囲条件を満たすことが保証されない。

[0101] そこで、次元 f に関する座標列 P_f に注目する。もし $P_f[s, e]$ に含まれる座標 $P_f[i]$ が、次元 f に関する範囲条件を満たすとき、座標 $P_f[i]$ に対応する点 $P[i]$ は、全ての次元に関する d 個の範囲条件を満たす。より厳密に表現すれば、 $s \leq i \leq e$ を満たす i に関して、座標 $P_f[i]$ が、次元 f における範囲条件 $l_{qf} \leq P_f[i] \leq u_{qf}$ をさらに満たすとき、この座標 $P_f[i]$ はクエリの範囲条件を全て満たす点 $P[i]$ に属する。

[0102] 本実施の形態では、このような性質が利用される。すなわち $P_f[s, e]$ に含まれる座標であって、座標の値が次元 f に関するクエリの範囲 $[l_{qf}, u_{qf}]$ に含まれるような全ての座標について、それらの座標が属する点の集合の統計量を計算する。この統計量は、ウェーブレット木を用いることで高速に計算できる。この統計量は、 $P[s, e]$ に含まれる点であって、クエリに含まれる点の集合の統計量に等しい。この統計量を、全ての区間について計算することで、クエリに含まれる全ての点の集合に関する全体的な統計量を求めることができる。

[0103] [装置動作]

次に、本発明の実施の形態における情報処理装置 100 の動作について図 6 を用いて説明する。図 6 は、本発明の実施の形態における情報処理装置の動作を示すフロー図である。また、以下の説明においては、適宜図 1 ～図 5 を参照する。また、本実施の形態では、情報処理装置 100 を動作させることによって、情報処理方法が実施される。よって、本実施の形態における情報処理方法の説明は、以下の情報処理装置 100 の動作説明に代える。

[0104] 図 6 に示すように、まず、入力受付部 50 は、クエリ領域の範囲を指定するための外部からの入力を受け付け（ステップ A 1）、受け付けた内容を区間検索部 10 に出力する。この入力されたクエリ領域 Q は、 $Q = [l_{q1}, u_{q1}] \times [l_{q2}, u_{q2}] \times \dots \times [l_{qd}, u_{qd}]$ と表記される。

[0105] 次に、区間検索部 10 は、統計量の集合を表す変数である AS に、空集合をセットする（ステップ A 2）。この変数 AS は、クエリに含まれる点の部分集合に関する部分的な統計量を、中間集計として格納するための変数である。

[0106] 次に、区間検索部 10 は、記憶部 43 に問い合わせを行なって、区間検索用データ構造 41、すなわち kd 木を取得する。区間検索部 10 は、 kd 木のルートノードを変数 v に代入する（ステップ A 3）。この変数 v は、現在注目しているノードを表す変数である。

[0107] 区間検索部 10 は、クエリ領域 Q について、区間検索用データ構造 41 を対象として関数 $\text{find_intervals}(v, Q)$ を実行し、区間と次元とのペアの集合 IDP を戻り値として取得する（ステップ A 4）。 IDP に含まれる区間 $I_v = [s, e]$ と次元 f とのペアは、点の列 P において、 $P[s, e]$ に含まれる $e - s + 1$ 個の点が、次元 f 以外の $(d - 1)$ 個の範囲条件を満たしていることを表している。また、関数 $\text{find_intervals}(v, Q)$ は、そのような IDP を返す関数である。

[0108] なお、関数 $\text{find_intervals}(v, Q)$ は、クエリに含まれる点であって、点の列 P 上でこれらの区間に含まれない点を見つけた場合、それらの点については別途統計量を計算して変数 AS に格納する。区間検索部 10 は、 IDP および AS を集計部 20 に出力する。

[0109] 次に、集計部 20 は、区間検索部 10 から IDP および AS を受け取り、 IDP に

含まれる区間 $I_v=[s, e]$ と次元 f とのペアについて、ループを開始する（ステップA5）。すなわち、IDPに含まれる全てのペアについて、ステップA6及びA7を実行する。

[0110] 次に、集計部20は、次元 f に関する座標列集計部30-fに区間 I_v を出力する。次元 f に関する座標列集計部30-fは、区間 I_v を入力として受け付け、記憶部43に問い合わせを行なって、次元 f の座標列 P_f に対応する座標列集計用データ構造42、すなわちウェーブレット木 w_f を取得する。そして、次元 f に関する座標列集計部30-fは、ウェーブレット木 w_f のルートノードを変数 v に代入する（ステップA6）。

[0111] 次元 f に関する座標列集計部30-fは、関数 $\text{aggregate_interval}(v, s, e, l_{qf}, u_{qf})$ を呼び出し、この関数が返す統計量（出力結果）をASに加える（ステップA7）。この関数 $\text{aggregate_interval}(v, s, e, l_{qf}, u_{qf})$ は、ウェーブレット木 w_f を参照することで実行される。また、関数 $\text{aggregate_interval}(v, s, e, l_{qf}, u_{qf})$ は、座標列 P_f に関して $P_f[s, e]$ に含まれる座標のうち、 $l_{qf} \leq P_f[i] \leq u_{qf}$ が成り立つ全ての座標 $P_f[i]$ の集合について、その集合に含まれる座標が属する全ての点の集合を特定し、その点の集合に関する統計量を返す関数である。これらの点はクエリに含まれる点の一部であり、この統計量は部分的な統計量である。

[0112] 統計量としては、条件を満たす点の数であるカウント(COUNT)、及び条件を満たす点の重みの合計であるサム(SUM)などを用いることができる。

[0113] 集計部20は、IDPに含まれる全てのペアについてステップA7が実行されたのち、ループを終了する（ステップA8）。

[0114] 集計部20は、ASに含まれる部分的な統計量を用いて、クエリ領域に含まれる全ての点の集合についての全体的な統計量を計算する（ステップA9）。たとえば、統計量としてカウントを用いる場合は、ASに含まれる各カウントを合計することにより、クエリ領域に含まれるすべての点の集合のカウントが得られる。

[0115] 最後に、出力部60は、集計部20から受け取った、クエリ領域に含まれ

る全ての点の集合についての全体的な統計量を外部に出力する（ステップA 10）。ステップA 1～ステップA 10の実行により、クエリ領域Qについての探索処理は終了する。また、ステップA 1～ステップA 10は、クエリ領域Qが入力される度に実行される。

[0116] [ステップA 4]

続いて、図6に示したステップA 4について、図7を用いて更に具体的に説明する。図7は、再帰的に区間を検索する関数find_intervals(v, Q)の動作を示すフロー図である。この関数は区間検索部10が、記憶部43に問い合わせを行なうことで実現される。

[0117] 図7に示すように、まず、区間検索部10は、kd木のノードvが葉ノードであるか判定する（ステップB 1）。そして、区間検索部10は、ステップB 1の判定の結果、答えがYesであれば、葉ノードに保持されている全ての点についてクエリ領域に含まれるか調べ、葉ノードに保持されており、かつクエリ領域に含まれる点について統計量を計算し、ASに加える（ステップB 6）。また、ステップB 6では、必要であれば、葉ノードに保持されている重みを参照して計算する（ステップB 6）。

[0118] また、図7においては、ステップB 6で行なわれる操作は、 $AS = AS \cup \text{aggregate_leaf}(v)$ という式で表現されている。aggregate_leaf(v)は、葉ノードに保持されている全ての点にクエリ領域に含まれているかどうかを調べ、クエリ領域に含まれる点について統計量を計算して返す関数である。たとえば、カウント・クエリを実現する場合、関数aggregate_leaf(v)は、葉ノードvに保持されている全ての点のうち、クエリ領域に含まれる点の数を数えて返す。以上の葉ノードに関する処理を行った場合、区間検索部10は空集合を返す。

[0119] 一方、ステップB 1の判定の結果、答えがNoであれば、区間検索部10は、kd木のノードvのカバー領域がクエリ領域と重なるかどうかを判定する（ステップB 2）。そして、区間検索部10は、ステップB 2の判定の結果、答えがYesであればステップB 3に進み、答えがNoであれば空集合を返す。

[0120] 具体的には、ステップB 2では、区間検索部10は、kd木のノードvのカバー領域 $R(v) = [l_{v1}, u_{v1}] \times [l_{v2}, u_{v2}] \times \dots \times [l_{vd}, u_{vd}]$ を取得する。その上で、区間検索部10は、 $1 \leq k \leq d$ となるkのうち、少なくともひとつの次元kにおいて、 $(u_{vk} < l_{qk}$ または $u_{qk} < l_{vk})$ が成り立つかどうかを判定する。判定の結果、上記関係が成り立つ場合は、空間上の重なりがないため、区間検索部10は、Noと判定する。上記の関係が成り立たない場合は、空間上での重なりがあるため、区間検索部10は、Yesと判定する。ステップB 2の判定は、クエリ領域と重なりがないカバー領域をこれ以上探索しないように枝刈りすることが目的である。

[0121] 次に、区間検索部10は、ノードvのカバー領域とクエリ領域とを比較して、包含次元数hを計算する（ステップB 3）。具体的には、区間検索部10は、たとえば、定義により、 $l_{qk} \leq l_{vk}$ 、且つ $u_{vk} \leq u_{qk}$ を満たす次元kがいくつ存在しているかを数えることによって、包含次元数hを計算できる。

[0122] 次に、区間検索部10は、包含次元数hがd-1よりも小さいかどうかを判定する（ステップB 4）。区間検索部10は、ステップB 4の判定の結果、答えがYesである場合は、区間検索部10は、変数 v_{left} にノードvの左の子ノードを代入し、変数 v_{right} にノードvの右の子ノードを代入する（ステップB 5）。

[0123] そして、ステップB 5の後、区間検索部10は、同じ関数を以下のように再帰的に呼び出す。

```
return find_intervals( $v_{left}$ , Q)  $\cup$  find_intervals( $v_{right}$ , Q)
```

[0124] 一方、ステップB 4の判定の結果、答えがNoである場合は、区間検索部10は、ノードvのカバー領域とクエリ領域とを比較して、 $l_{qf} \leq l_{vf}$ 、且つ $u_{vf} \leq u_{qf}$ という範囲条件が満たされていない次元fを求める（ステップB 7）。そして、区間検索部10は、ノードvが保持する添字の区間 I_v と次元fとのペアである (I_v, f) を返す。

[0125] 以上により、図7に示すアルゴリズムの動作の説明を終える。この図7に示すアルゴリズムは、従来のkd木の探索アルゴリズムとほぼ等しいが、d個の範囲条件全てを満たすノードを見つけるまで探索するのではなく、(d-1)個の

範囲条件を満たすノードを見つけるまで探索する点で異なっている。

[0126] [ステップA 7]

次いで、図6に示すアルゴリズムにおけるステップA 7の動作について、図8を用いて詳しく説明する。すなわち、図6中の関数 $\text{aggregate_interval}(v, s, e, l_{qf}, u_{qf})$ の動作について、図8を用いて説明する。図8は、図6のステップA 7に示された関数 $\text{aggregate_interval}(v, s, e, l_{qf}, u_{qf})$ の動作を示す図である。

[0127] 関数 $\text{aggregate_interval}(v, s, e, l_{qf}, u_{qf})$ は、次元 f に関する座標列集計部30-fによって実行される関数である。この関数は、ウェーブレット木 w_f のノード v と、添字の区間 $[s, e]$ 、および、座標の値の範囲 $[l_{qf}, u_{qf}]$ を入力として受け付け、 v に対応する座標部分列 $P_f(\pi)$ 上で該区間に含まれる座標のうち、座標の値が該範囲に含まれる全ての座標について、それらの座標が属する点に関する統計量を返す関数である。

[0128] 図8に示すように、次元 f に関する座標列集計部30-fは、関数 $\text{aggregate_interval}(v, s, e, l_{qf}, u_{qf})$ を実行して、 $s > e$ または $([l_{\pi}, u_{\pi}] \cap [l_{qf}, u_{qf}]) = \phi$ が成り立つかどうかを判定する(ステップC 1)。そして、座標列集計部30-fは、ステップC 1の判定の結果、答えがYesであれば、空集合を返す。なお、 $[l_{\pi}, u_{\pi}]$ は、接頭辞 π で始まる整数の範囲を表している。

[0129] 一方、ステップC 1の判定の結果、答えがNoであれば、座標列集計部は、 $[l_{\pi}, u_{\pi}] \subseteq [l_{qf}, u_{qf}]$ が成り立つかどうかを判定する(ステップC 2)。なお、 $[l_{\pi}, u_{\pi}]$ は、接頭辞 π で始まる整数の範囲を表している。

[0130] そして、ステップC 2の判定の結果、答えがYesであれば、つまり、 $[l_{\pi}, u_{\pi}] \subseteq [l_{qf}, u_{qf}]$ が成り立つときは、座標の値の範囲がクエリの範囲に包含される。よって、 $P_f(\pi)[s, e]$ に含まれる座標は、必ずクエリに包含される点に属する。従って、座標列集計部30-fは、関数 $\text{aggregate_node}(v, s, e)$ を実行して、その出力結果を返り値とする。関数 $\text{aggregate_node}(v, s, e)$ は、 v に対応する座標列 $P_f(\pi)$ に関して、 $P_f(\pi)[s, e]$ に含まれる座標が属する点の集合の統計量を返す関数である。

[0131] 一方、ステップC 2の判定の結果、答えがNoであれば、座標列集計部30は、ノードvに保持されているビット列を B_v とにおいて、図8中に示す4つのrankの式を用いて、左の子ノードにおける添字の区間 $[s_{\text{left}}, e_{\text{left}}]$ と、右の子ノードにおける添字の区間 $[s_{\text{right}}, e_{\text{right}}]$ とを計算する（ステップC 3）。

[0132] これらの式を用いて計算すると、ウェーブレット木の特性により、ノードvに対応する座標部分列 $P_f(\pi)$ 上の区間 $[s, e]$ から抽出された座標を含む、左の子ノードに対応する座標部分列 $P_f(\pi_{\text{left}})$ 上の区間 $[s_{\text{left}}, e_{\text{left}}]$ 、および、右の子ノードに対応する座標部分列 $P_f(\pi_{\text{right}})$ 上の区間 $[s_{\text{right}}, e_{\text{right}}]$ を計算できる。なお、 π_{left} 、 π_{right} は接頭辞 π を一つ延長したものであり、 π_{left} は $\pi + "0"$ に対応し、 π_{right} は $\pi + "1"$ に対応している。

[0133] その後、座標列集計部30は、右の子ノード、および左の子ノードについて、同様の処理を行うため、再帰的に以下の関数を呼び出す。

```
return aggregate_interval (vleft, sleft, eleft, lqf, uqf) ∪ aggregate_interval (vright, sright, eright, lqf, uqf)
```

[0134] [ステップC 2]

続いて、図8に示したステップC 2で呼び出される関数 $\text{aggregate_node}(v, s, e)$ について説明する。この関数は座標列集計部30によって実行される。

[0135] 関数 $\text{aggregate_node}(v, s, e)$ は、ノードvに対応する座標列 $P_f(\pi)$ に関して、 $P_f(\pi)[s, e]$ に含まれる座標が属する点の集合の統計量を返す関数である。

[0136] 関数 $\text{aggregate_node}(v, s, e)$ は、様々な集計関数を抽象化したものであり、この関数を、具体的な集計関数に置き換えることで、情報処理装置100を様々な種類の矩形範囲検索に利用できる。

[0137] たとえば、情報処理装置100は、クエリ領域Qに含まれる点の数をカウントして出力することができる。この動作は、関数 $\text{aggregate_node}(v, s, e)$ が、返り値として $(e - s + 1)$ を返すことによって実現される。何故なら、 $P_f(\pi)[s, e]$ に含まれる全ての座標が、それぞれクエリ領域Qに含まれる点に対

応しており、 $(e - s + 1)$ 個の点がクエリ領域 Q に含まれていることを表しているからである。

[0138] また、このとき、関数 $\text{aggregate_interval}(v, s, e, l_{qf}, u_{qf})$ は、 $P_f[s, e]$ に座標が含まれる点のうち、さらにクエリ領域 Q に含まれる数をカウントして返す関数として動作する。この場合、集計部20は、点の列 P に含まれる点のうち、クエリ領域 Q に含まれる数をカウントして出力する。

[0139] また、たとえば、全ての点 p に重み $w(p)$ が付与されているとき、情報処理装置100は、クエリ領域に含まれる点の重みの合計を計算することができる。これは、あらかじめ、全ての座標部分列 $P_f(\pi)$ に含まれる各座標について、対応する点 p の重み $w(p)$ を同じ順番で並べた列 $W_f(\pi)$ が設定されている場合に、この列上での区間合計を計算できるようなデータ構造が用意されていれば可能となる。

[0140] このようなデータ構造としては、既存のPartial Sumを扱うデータ構造が挙げられる。また、このようなデータ構造であれば、 $P_f(\pi)[s, e]$ がクエリ領域に含まれる点に対応していることが分かっているとき、この区間 $[s, e]$ に対応する重みの列上の区間 $W_f(\pi)[s, e]$ に含まれる重みの区間合計を計算し、最後に足し合わせることで、クエリ領域 Q に含まれる全ての点の重みの合計が計算できる。この場合、集計部20は、クエリ領域 Q に含まれる全ての点の重みの合計を統計量として出力する。

[0141] 同様に、情報処理装置100は、クエリ領域 Q に含まれる全ての点のリストを返すレポート・クエリとしても用いることができる。つまり、 $P_f(\pi)[s, e]$ に含まれる各要素 $P_f(\pi)[j]$ について、元の整数列 P_f 上での位置 i を、ウェーブレット木をさかのぼることによって特定できる。このとき、点 $P[i]$ がクエリ領域に含まれる。この場合、集計部20は、クエリ領域 Q に含まれる全ての点のリストを統計量として出力する。

[0142] 以上で、関数 $\text{aggregate_interval}(v, s, e, l_{qf}, u_{qf})$ 及び関数 $\text{aggregate_node}(v, s, e)$ についての説明を終える。この2つの関数の動作は、非特許文献2に示されている、ウェーブレット木を用いた2次元上の統計量の計算

と同等であることに注意する。すなわち、この2つの関数の動作は、添字の区間 $[s, e]$ と値の範囲 $[l_{qf}, u_{qf}]$ を指定した2次元空間上での検索と見なすことができる。この計算によって求まる区間の数は $O(\log n)$ であることが知られている。

[0143] また、以上のように、本実施の形態によれば、様々な種類の矩形範囲検索を実現することができる。本実施の形態は、図6～図8に示したアルゴリズムが単独で用いられる態様に限定されず、図6～図8に示したアルゴリズムに、適宜他の探索アルゴリズムが組み合わされた態様であってもよい。

[0144] [実施の形態による効果]

本実施の形態は、kd木を単独で用いる従来手法よりも、計算量が少なくなる効果がある。このことを明らかにするため、最悪計算量について解析する。kd木を用いる従来手法は、包含次元数が d になるまで分割するのに対して、本実施の形態における手法は、 $d-1$ になるまでしか分割しない。このことが最悪計算量におよぼす影響を以下に述べる。

[0145] まず、計算量が最悪になる場合のkd木のノードの分割数を見積もる。計算量が最悪になるのは、空間分割数が最大になるときである。つまり、1回の分割で生じた2つのカバー領域が、常にクエリ領域と重なってしまう場合である。

[0146] 図9に、最悪な場合の探索ノード数と包含次元数との関係を示す。図9は、2次元の場合の探索ノード数と包含次元数との変化を示す図である。図9に示すように、木構造上のひとつのノードは、ひとつの探索ノードに対応している。木構造の深さが1つ下がることは、ノードが1回分割されて2つの探索ノードに分割されることを表している。ノード上の数字は、包含次元数を表している。分割されるほど、包含次元数が高いノードが増えていくことが分かる。

[0147] ここで、 d 回の分割をまとめて考える。深さ $m * d$ において包含次元数 h になるノードの数を $T_h(m)$ とにおいて、 $T_h(m)$ と $T_h(m-1)$ のあいだに成り立つ漸化式を考える。 d 回の分割により、1つのカバー領域が 2^d 個のカバー領域に分割される。

このとき、必ず各次元で1回の分割が生じる。すでに包含されている次元について分割されても包含次元数は上がらないことを考えると、深さ $m * d$ において、包含次元数が h になるノードの数を求めるには、深さ $(m-1) * d$ において包含次元数が $i (\leq h)$ だったノードから、 $h-i$ 個の次元が新たに包含される数を考えればよい。

[0148] この漸化式は、以下の数1に示す通りとなる。ただし、以下の数1において、 $C(x, y)$ は組み合わせの数を表すものとする。

[0149] [数1]

$$\begin{aligned} T_h(m) &= \sum_{i=0}^h 2^i C(d-i, h-i) T_i(m-1) \\ &= 2^h T_h(m-1) - 2^{h-1} (d-h+1) T_{h-1}(m-1) + \dots + T_0(m-1) \end{aligned}$$

[0150] 上記の数1から、 d 回の分割により、全体のノード数は 2^d 倍に増加するが、そのうち包含次元数 h のノードは 2^h 倍に増加することが分かる。

[0151] また、この分割を $\log(n)/d$ 回繰り返すと、探索木全体は深さ $\log n$ の二分木になり、全体のノード数は $O(n)$ に達して分割が終了する。そのうち包含次元数 h のノードは、 $O(n^{(h/d)})$ となる。ただし包含次元数 0 のノードは $O(\log n)$ である。

[0152] よって、以下のように説明することができる。まず、探索の打ち切りを全く行なわなければ、分割数は最大で $O(n)$ になる。また、包含次元数が d に達した時点で分割を打ち切れば、分割数は $O(n^{(d-1)/d})$ となる。一方、包含次元数が $d-1$ に達した時点で分割を打ち切れば、分割数は $O(n^{(d-2)/d})$ となる。そして、 kd 木では、包含次元数が d に達した時点で分割が打ち切られるため、計算量は $O(n^{(d-1)/d})$ となる。これは従来から知られているオーダと一致する。

[0153] この kd 木での解析が、本実施の形態に適用される。本実施の形態では、包含次元数が $d-1$ に達した時点で分割が打ち切られるので、分割数、すなわち kd 木によって計算される区間の数は最大で $O(n^{(d-2)/d})$ となる。

[0154] また、区間毎に、関数 $\text{aggregate_interval}(v, s, e, l_{qf}, u_{qf})$ が実行される。この関数では、関数 $\text{aggregate_node}(v, s, e)$ が $O(\log n)$ 回実行され

る。ここで関数`aggregate_node(v, s, e)`が $O(1)$ で実行できる関数だとする。たとえば、カウント・クエリを実現するには、 $(e - s + 1)$ を計算するだけでよい。そのため、 $O(1)$ で計算できる。これにより、本実施の形態における手法は、 $O(n^{(d-2)/d})$ 個の区間について、それぞれ $O(\log n)$ 回、 $O(1)$ の計算を実行することになるため、合計の計算量は $O(n^{(d-2)/d} \log n)$ となる。

[0155] ただし、 $d=2$ の場合は特殊である。 $d-1=1$ 次元が包含された時点で探索ループを抜けるので、分割されたノードの数は、包括次元数が0となるノードの数 $O(\log n)$ に比例する。各ノードについて $O(\log n)$ の計算が必要になるので、 $d=2$ の場合の計算量は $O(\log^2 n)$ となる。

[0156] 以上はクエリ領域に包含される点を数えるカウント・クエリの場合であるが、包含される点全てについてリストを出力するレポート・クエリには、出力される点の数を F として、それぞれについて $O(\log n)$ の計算時間がかかる。まとめると、図10に示す通りとなる。図10に示すように、本発明によれば、kd木を用いて探索処理を行なう場合よりも計算量のオーダーが改善されており、しかも従来のウェーブレット木と異なり、3次元以上にも適用できる。図10は、本発明と従来手法との計算量の比較を示す図である。

[0157] [プログラム]

本発明の実施の形態におけるプログラムは、コンピュータに、図6に示すステップA1～A10を実行させるプログラムであれば良い。このプログラムをコンピュータにインストールし、実行することによって、本実施の形態における情報処理装置100と情報処理方法とを実現することができる。この場合、コンピュータのCPU (Central Processing Unit) は、区間検索部10、集計部20、座標列集計部30、入力受付部50、及び出力部60として機能し、処理を行なう。また、本実施の形態では、記憶部43は、コンピュータに備えられたハードディスク等の記憶装置に、これらを構成するデータファイルを格納することによって実現される。

[0158] なお、本実施の形態におけるプログラムは、複数のコンピュータによって構築されたコンピュータシステムによって実行されても良い。この場合は、

例えば、各コンピュータが、それぞれ、検索部 10、集計部 20、座標列集計部 30、入力受付部 50、及び出力部 60 として機能しても良い。また、記憶部 43 は、本実施の形態におけるプログラムを実行するコンピュータとは別のコンピュータ上に構築されていても良い。

[0159] ここで、本実施の形態におけるプログラムを実行することによって、情報処理装置 100 を実現するコンピュータについて図 11 を用いて説明する。図 11 は、本発明の実施の形態における情報処理装置を実現するコンピュータの一例を示すブロック図である。

[0160] 図 11 に示すように、コンピュータ 110 は、CPU 111 と、メインメモリ 112 と、記憶装置 113 と、入力インターフェイス 114 と、表示コントローラ 115 と、データリーダ／ライタ 116 と、通信インターフェイス 117 とを備える。これらの各部は、バス 121 を介して、互いにデータ通信可能に接続される。

[0161] CPU 111 は、記憶装置 113 に格納された、本実施の形態におけるプログラム（コード）をメインメモリ 112 に展開し、これらを所定順序で実行することにより、各種の演算を実施する。メインメモリ 112 は、典型的には、DRAM（Dynamic Random Access Memory）等の揮発性の記憶装置である。また、本実施の形態におけるプログラムは、コンピュータ読み取り可能な記録媒体 120 に格納された状態で提供される。なお、本実施の形態におけるプログラムは、通信インターフェイス 117 を介して接続されたインターネット上で流通するものであっても良い。

[0162] また、記憶装置 113 の具体例としては、ハードディスクドライブの他、フラッシュメモリ等の半導体記憶装置が挙げられる。入力インターフェイス 114 は、CPU 111 と、キーボード及びマウスといった入力機器 118 との間のデータ伝送を仲介する。表示コントローラ 115 は、ディスプレイ装置 119 と接続され、ディスプレイ装置 119 での表示を制御する。

[0163] データリーダ／ライタ 116 は、CPU 111 と記録媒体 120 との間のデータ伝送を仲介し、記録媒体 120 からのプログラムの読み出し、及びコ

ンピュータ 110 における処理結果の記録媒体 120 への書き込みを実行する。通信インターフェイス 117 は、CPU 111 と、他のコンピュータとの間のデータ伝送を仲介する。

[0164] また、記録媒体 120 の具体例としては、CF (Compact Flash (登録商標)) 及びSD (Secure Digital) 等の汎用的な半導体記憶デバイス、フレキシブルディスク (Flexible Disk) 等の磁気記憶媒体、又はCD-ROM (Compact Disk Read Only Memory) などの光学記憶媒体が挙げられる。

[0165] また、上述した実施の形態の一部又は全部は、以下に記載する(付記1)～(付記28)によって表現することができるが、以下の記載に限定されるものではない。

[0166] (付記1) 多次元空間上の点の集合を表現するデータ構造を処理対象とする情報処理装置であって、

クエリ領域として、特定の多次元の領域が指定された場合に、前記点の集合を一行に並べて得られた点の列上にあり、且つ、前記多次元空間を構成する全次元のうち1つの次元を除いた残りの各次元の座標がクエリ領域に含まれている点のみによって構成されている、区間を特定する、区間検索部と、

前記区間検索部で特定された区間について、当該区間に出現する点が前記クエリ領域に含まれるための条件として、除かれた前記1つの次元における座標の値の範囲を特定する、集計部と、

前記区間検索部で特定された区間、及び前記集計部で特定された前記座標の値の範囲を入力として、

前記点の列の並び順と同じ順序で、前記点の集合の各点における、除かれた前記1つの次元での座標を取り出すことで得られる、座標列に関して、当該座標列において入力された前記区間に出現し、且つ、値が入力された前記範囲に含まれる、全ての座標について、

前記全ての座標が対応する点の集合に関する統計量を計算する、座標列集計部と、

を備えている、ことを特徴とする情報処理装置。

[0167] (付記2) 前記座標列集計部が、前記多次元空間を構成する全次元それぞれ毎に備えられており、それぞれ、対応する次元と前記集計部が座標の値の範囲を特定した次元とが一致する場合に、前記点の集合に関する統計量を計算する、

付記1に記載の情報処理装置。

[0168] (付記3) 前記集計部が、前記区間検索部によって特定された区間が複数である場合に、前記座標列集計部によって計算された区間毎の前記点の集合に関する統計量を、更に集計し、集計によって得られた統計量を、前記クエリ領域に含まれる点の集合に関する全体的な統計量として、出力する、

付記1に記載の情報処理装置。

[0169] (付記4) 前記データ構造が、前記区間検索部による前記区間の特定に用いられる第1のデータ構造と、前記座標列集計部による前記統計量の計算に用いられる第2のデータ構造とを有している、

付記1に記載の情報処理装置。

[0170] (付記5) 前記第1のデータ構造が、
前記多次元空間に設定された複数のカバー領域のいずれかと、前記点の列上において当該カバー領域に含まれる点が出現する区間とに関連付けられている、ノードを有する木構造によって表現されており、

前記区間検索部が、

前記ノードのうち、

関連付けられた前記カバー領域に含まれる点における、前記1つの次元を除いた残りの各次元の座標がクエリ領域に含まれている、ノードを特定し、
特定した1又は2以上のノードが関連付けられている区間を前記区間として特定する、

付記4に記載の情報処理装置。

[0171] (付記6) 前記点の列は、

前記ノードに関連付けられたカバー領域それぞれに含まれる点が、ひとつな

がりで連続して出現するように、前記点の集合に含まれる点を一列に並べて得られている、

付記 5 に記載の情報処理装置。

[0172] (付記 7) 前記座標列集計部は、前記第 2 のデータ構造を用いて、前記座標列から得られる複数の部分列のうち、入力された前記範囲に含まれる座標のみが出現する部分列を特定し、そして、特定した部分列上の区間であって、前記座標列において入力された前記区間に出現する座標が出現している、第二の区間を特定し、

更に、特定した前記第二の区間に出現する座標が対応する点の集合に関する統計量を計算する、

付記 4 に記載の情報処理装置。

[0173] (付記 8) 前記部分列が、座標のビット表現が同じ接頭辞で始まる座標を、座標同士の位置関係を保ったまま抽出することで得られるものであり、

前記第 2 のデータ構造が、

前記部分列に関連付けられた複数のノードを有し、

前記複数のノードそれぞれは、前記部分列において出現する各座標のビット表現における、一つ以上の特定の桁のビットを取り出し、取り出した前記ビットを前記部分列と同じ順序で並べることによって得られる、ビットの列を用いて表現され、

前記座標列集計部は、前記複数ノードそれぞれを表現するビットの列を用いて、前記第二の区間を特定する、

付記 7 に記載の情報処理装置。

[0174] (付記 9) 前記座標列集計部が、前記全ての座標が対応する点の集合に関する統計量として、前記全ての座標が対応する点の個数を計算する、

付記 1 に記載の情報処理装置。

[0175] (付記 10) 前記座標列集計部が、前記全ての座標が対応する点の集合に関する統計量として、前記全ての座標が対応する点それぞれの各次元の座標を計算する、

付記 1 に記載の情報処理装置。

[0176] (付記 1 1) 多次元空間上の点の集合を表現するデータ構造を処理対象とする情報処理方法であって、

(a) クエリ領域として、特定の多次元の領域が指定された場合に、前記点の集合を一行に並べて得られた点の列上にあり、且つ、前記多次元空間を構成する全次元のうち 1 つの次元を除いた残りの各次元の座標がクエリ領域に含まれている点のみによって構成されている、区間を特定する、ステップと、

(b) 前記 (a) のステップで特定された区間について、当該区間に出現する点が前記クエリ領域に含まれるための条件として、除かれた前記 1 つの次元における座標の値の範囲を特定する、ステップと、

(c) 前記 (a) のステップで特定された区間、及び前記 (b) のステップで特定された前記座標の値の範囲を入力として、

前記点の列の並び順と同じ順序で、前記点の集合の各点における、除かれた前記 1 つの次元での座標を取り出すことで得られる、座標列に関して、当該座標列において入力された前記区間に出現し、且つ、値が入力された前記範囲に含まれる、全ての座標について、

前記全ての座標が対応する点の集合に関する統計量を計算する、ステップと、

を有する、ことを特徴とする情報処理方法。

[0177] (付記 1 2) (d) 前記 (a) のステップによって特定された区間が複数である場合に、前記 (b) のステップによって計算された区間毎の前記点の集合に関する統計量を、更に集計し、集計によって得られた統計量を、前記クエリ領域に含まれる点の集合に関する全体的な統計量として、出力する、ステップを更に有する、

付記 1 1 に記載の情報処理方法。

[0178] (付記 1 3) 前記データ構造が、前記 (a) のステップによる前記区間の特定に用いられる第 1 のデータ構造と、前記 (c) のステップによる前記統

計量の計算に用いられる第2のデータ構造とを有している、
付記11に記載の情報処理方法。

- [0179] (付記14) 前記第1のデータ構造が、
前記多次元空間に設定された複数のカバー領域のいずれかと、前記点の列上
において当該カバー領域に含まれる点が出現する区間とに関連付けられてい
る、ノードを有する木構造によって表現されており、
前記(a)のステップにおいて、
前記ノードのうち、
関連付けられた前記カバー領域に存在する点における、前記1つの次元を除
いた残りの各次元の座標がクエリ領域に含まれている、ノードを特定し、
特定した1又は2以上のノードが関連付けられている区間を前記区間として
特定する、
付記13に記載の情報処理方法。

- [0180] (付記15) 前記点の列は、
前記ノードに関連付けられたカバー領域それぞれに存在する点が、ひとつな
がりで連続して出現するように、前記点の集合に含まれる点を一列に並べて
得られている、
付記14に記載の情報処理方法。

- [0181] (付記16) 前記(c)のステップにおいて、前記第2のデータ構造を用
いて、
前記座標列から得られる複数の部分列のうち、入力された前記範囲に含まれ
る座標のみが出現する部分列を特定し、そして、特定した部分列上の区間で
あって、前記座標列において入力された前記区間に出現する座標が出現して
いる、第二の区間を特定し、
更に、特定した前記第二の区間に出現する座標が対応する点の集合に関する
統計量を計算する、
付記13に記載の情報処理方法。

- [0182] (付記17) 前記部分列が、座標のビット表現が同じ接頭辞で始まる座標

を、座標同士の位置関係を保ったまま抽出することで得られるものであり、

前記第2のデータ構造が、

前記部分列に関連付けられた複数のノードを有し、

前記複数のノードそれぞれは、前記部分列において出現する各座標のビット表現における、一つ以上の特定の桁のビットを取り出し、取り出した前記ビットを前記部分列と同じ順序で並べることによって得られる、ビットの列を用いて表現され、

前記(c)のステップにおいて、前記複数ノードそれぞれを表現するビットの列を用いて、前記第二の区間を特定する、

付記16に記載の情報処理方法。

[0183] (付記18) 前記(c)のステップにおいて、前記全ての座標が対応する点の集合に関する統計量として、前記全ての座標が対応する点の個数を計算する、

付記11に記載の情報処理方法。

[0184] (付記19) 前記(c)のステップにおいて、前記全ての座標が対応する点の集合に関する統計量として、前記全ての座標が対応する点それぞれの各次元の座標を計算する、

付記11に記載の情報処理方法。

[0185] (付記20) 多次元空間上の点の集合を表現するデータ構造を処理対象とする情報処理をコンピュータによって行なうためのプログラムを記録したコンピュータ読み取り可能な記録媒体であって、

前記コンピュータに、

(a) クエリ領域として、特定の多次元の領域が指定された場合に、前記点の集合を一行に並べて得られた点の列上にあり、且つ、前記多次元空間を構成する全次元のうち1つの次元を除いた残りの各次元の座標がクエリ領域に含まれている点のみによって構成されている、区間を特定する、ステップと、

(b) 前記(a)のステップで特定された区間について、当該区間に出現す

る点が前記クエリ領域に含まれるための条件として、除かれた前記1つの次元における座標の値の範囲を特定する、ステップと、

(c) 前記(a)のステップで特定された区間、及び前記(b)のステップで特定された前記座標の値の範囲を入力として、

前記点の列の並び順と同じ順序で、前記点の集合の各点における、除かれた前記1つの次元での座標を取り出すことで得られる、座標列に関して、当該座標列において入力された前記区間に出現し、且つ、値が入力された前記範囲に含まれる、全ての座標について、

前記全ての座標が対応する点の集合に関する統計量を計算する、ステップと、

、
を実行させる命令を含む、プログラムを記録しているコンピュータ読み取り可能な記録媒体。

[0186] (付記21) 前記プログラムが、前記コンピュータに、

(d) 前記(a)のステップによって特定された区間が複数である場合に、前記(b)のステップによって計算された区間毎の前記点の集合に関する統計量を、更に集計し、集計によって得られた統計量を、前記クエリ領域に含まれる点の集合に関する全体的な統計量として、出力する、ステップを実行させる命令を更に含む、

付記20に記載のコンピュータ読み取り可能な記録媒体。

[0187] (付記22) 前記データ構造が、前記(a)のステップによる前記区間の特定に用いられる第1のデータ構造と、前記(c)のステップによる前記統計量の計算に用いられる第2のデータ構造とを有している、

付記20に記載のコンピュータ読み取り可能な記録媒体。

[0188] (付記23) 前記第1のデータ構造が、

前記多次元空間に設定された複数のカバー領域のいずれかと、前記点の列上において当該カバー領域に含まれる点が出現する区間とに関連付けられている、ノードを有する木構造によって表現されており、

前記(a)のステップにおいて、

前記ノードのうち、

関連付けられた前記カバー領域に存在する点における、前記 1 つの次元を除いた残りの各次元の座標がクエリ領域に含まれている、ノードを特定し、
特定した 1 又は 2 以上のノードが関連付けられている区間を前記区間として
特定する、

付記 2 2 に記載のコンピュータ読み取り可能な記録媒体。

[0189] (付記 2 4) 前記点の列は、

前記ノードに関連付けられたカバー領域それぞれに存在する点が、ひとつながりで連続して出現するように、前記点の集合に含まれる点を一列に並べて
得られている、

付記 2 3 に記載のコンピュータ読み取り可能な記録媒体。

[0190] (付記 2 5) 前記 (c) のステップにおいて、前記第 2 のデータ構造を用
いて、

前記座標列から得られる複数の部分列のうち、入力された前記範囲に含まれ
る座標のみが出現する部分列を特定し、そして、特定した部分列上の区間で
あって、前記座標列において入力された前記区間に出現する座標が出現して
いる、第二の区間を特定し、

更に、特定した前記第二の区間に出現する座標が対応する点の集合に関する
統計量を計算する、

付記 2 2 に記載のコンピュータ読み取り可能な記録媒体。

[0191] (付記 2 6) 前記部分列が、座標のビット表現が同じ接頭辞で始まる座標
を、座標同士の位置関係を保ったまま抽出することで得られるものであり、
前記第 2 のデータ構造が、

前記部分列に関連付けられた複数のノードを有し、

前記複数のノードそれぞれは、前記部分列において出現する各座標のビット
表現における、一つ以上の特定の桁のビットを取り出し、取り出した前記ビ
ットを前記部分列と同じ順序で並べることによって得られる、ビットの列を
用いて表現され、

前記（c）のステップにおいて、前記複数ノードそれぞれを表現するビットの列を用いて、前記第二の区間を特定する、

付記 25 に記載のコンピュータ読み取り可能な記録媒体。

[0192] （付記 27） 前記（c）のステップにおいて、前記全ての座標が対応する点の集合に関する統計量として、前記全ての座標が対応する点の個数を計算する、

付記 20 に記載のコンピュータ読み取り可能な記録媒体。

[0193] （付記 28） 前記（c）のステップにおいて、前記全ての座標が対応する点の集合に関する統計量として、前記全ての座標が対応する点それぞれの各次元の座標を計算する、

付記 20 に記載のコンピュータ読み取り可能な記録媒体。

[0194] 以上、実施の形態を参照して本願発明を説明したが、本願発明は上記実施の形態に限定されるものではない。本願発明の構成や詳細には、本願発明のスコープ内で当業者が理解し得る様々な変更をすることができる。

[0195] この出願は、2014年11月7日に提出された日本出願特願2014-227041を基礎とする優先権を主張し、その開示の全てをここに取り込む。

産業上の利用可能性

[0196] 以上のように、本発明によれば、任意の次元dについて、線形サイズで、kd木よりも高速な矩形範囲検索を実現することができる。本発明は、大量のデータ群から必要なデータを探索する必要がある種々の分野において有用である。

符号の説明

[0197] 10 区間検索部
 20 集計部
 30、30-1～30-d 座標列集計部
 40 データ構造
 41 区間検索用データ構造

- 4 2 座標列集計用データ構造
- 4 3 記憶部
- 5 0 入力受付部
- 6 0 出力部
- 1 0 0 情報処理装置
- 1 1 0 コンピュータ
- 1 1 1 C P U
- 1 1 2 メインメモリ
- 1 1 3 記憶装置
- 1 1 4 入力インターフェイス
- 1 1 5 表示コントローラ
- 1 1 6 データリーダ／ライタ
- 1 1 7 通信インターフェイス
- 1 1 8 入力機器
- 1 1 9 ディスプレイ装置
- 1 2 0 記録媒体
- 1 2 1 バス

請求の範囲

[請求項1] 多次元空間上の点の集合を表現するデータ構造を処理対象とする情報処理装置であって、

クエリ領域として、特定の多次元の領域が指定された場合に、前記点の集合を一行に並べて得られた点の列上にあり、且つ、前記多次元空間を構成する全次元のうち1つの次元を除いた残りの各次元の座標がクエリ領域に含まれている点のみによって構成されている、区間を特定する、区間検索部と、

前記区間検索部で特定された区間について、当該区間に出現する点の前記クエリ領域に含まれるための条件として、除かれた前記1つの次元における座標の値の範囲を特定する、集計部と、

前記区間検索部で特定された区間、及び前記集計部で特定された前記座標の値の範囲を入力として、

前記点の列の並び順と同じ順序で、前記点の集合の各点における、除かれた前記1つの次元での座標を取り出すことで得られる、座標列に関して、当該座標列において入力された前記区間に出現し、且つ、値が入力された前記範囲に含まれる、全ての座標について、前記全ての座標が対応する点の集合に関する統計量を計算する、座標列集計部と、

を備えている、ことを特徴とする情報処理装置。

[請求項2] 前記座標列集計部が、前記多次元空間を構成する全次元それぞれ毎に備えられており、それぞれ、対応する次元と前記集計部が座標の値の範囲を特定した次元とが一致する場合に、前記点の集合に関する統計量を計算する、

請求項1に記載の情報処理装置。

[請求項3] 前記集計部が、前記区間検索部によって特定された区間が複数である場合に、前記座標列集計部によって計算された区間毎の前記点の集合に関する統計量を、更に集計し、集計によって得られた統計量を、

前記クエリ領域に含まれる点の集合に関する全体的な統計量として、
出力する、

請求項 1 または 2 に記載の情報処理装置。

[請求項4] 前記データ構造が、前記区間検索部による前記区間の特定に用いられる第 1 のデータ構造と、前記座標列集計部による前記統計量の計算に用いられる第 2 のデータ構造とを有している、
請求項 1 ～ 3 のいずれかに記載の情報処理装置。

[請求項5] 前記第 1 のデータ構造が、
前記多次元空間に設定された複数のカバー領域のいずれかと、前記点の列上において当該カバー領域に含まれる点が出現する区間とに関連付けられている、ノードを有する木構造によって表現されており、
前記区間検索部が、
前記ノードのうち、
関連付けられた前記カバー領域に含まれる点における、前記 1 つの次元を除いた残りの各次元の座標がクエリ領域に含まれている、ノードを特定し、
特定した 1 又は 2 以上のノードが関連付けられている区間を前記区間として特定する、
請求項 4 に記載の情報処理装置。

[請求項6] 前記点の列は、
前記ノードに関連付けられたカバー領域それぞれに含まれる点が、ひとつつながりで連続して出現するように、前記点の集合に含まれる点を
一列に並べて得られている、
請求項 5 に記載の情報処理装置。

[請求項7] 前記座標列集計部は、前記第 2 のデータ構造を用いて、
前記座標列から得られる複数の部分列のうち、入力された前記範囲に含まれる座標のみが出現する部分列を特定し、そして、特定した部分列上の区間であって、前記座標列において入力された前記区間に出現

する座標が出現している、第二の区間を特定し、
更に、特定した前記第二の区間に出現する座標が対応する点の集合に関する統計量を計算する、
請求項 4 ～ 6 のいずれかに記載の情報処理装置。

[請求項 8] 前記部分列が、座標のビット表現が同じ接頭辞で始まる座標を、座標同士の位置関係を保ったまま抽出することで得られるものであり、
前記第 2 のデータ構造が、
前記部分列に関連付けられた複数のノードを有し、
前記複数のノードそれぞれは、前記部分列において出現する各座標のビット表現における、一つ以上の特定の桁のビットを取り出し、取り出した前記ビットを前記部分列と同じ順序で並べることによって得られる、ビットの列を用いて表現され、
前記座標列集計部は、前記複数ノードそれぞれを表現するビットの列を用いて、前記第二の区間を特定する、
請求項 7 に記載の情報処理装置。

[請求項 9] 前記座標列集計部が、前記全ての座標が対応する点の集合に関する統計量として、前記全ての座標が対応する点の個数を計算する、
請求項 1 ～ 8 のいずれかに記載の情報処理装置。

[請求項 10] 前記座標列集計部が、前記全ての座標が対応する点の集合に関する統計量として、前記全ての座標が対応する点それぞれの各次元の座標を計算する、
請求項 1 ～ 8 のいずれかに記載の情報処理装置。

[請求項 11] 多次元空間上の点の集合を表現するデータ構造を処理対象とする情報処理方法であって、
(a) クエリ領域として、特定の多次元の領域が指定された場合に、前記点の集合を一行に並べて得られた点の列上にあり、且つ、前記多次元空間を構成する全次元のうち 1 つの次元を除いた残りの各次元の座標がクエリ領域に含まれている点のみによって構成されている、区

間を特定する、ステップと、

(b) 前記 (a) のステップで特定された区間について、当該区間に出現する点が前記クエリ領域に含まれるための条件として、除かれた前記 1 つの次元における座標の値の範囲を特定する、ステップと、

(c) 前記 (a) のステップで特定された区間、及び前記 (b) のステップで特定された前記座標の値の範囲を入力として、
前記点の列の並び順と同じ順序で、前記点の集合の各点における、除かれた前記 1 つの次元での座標を取り出すことで得られる、座標列に関して、当該座標列において入力された前記区間に出現し、且つ、値が入力された前記範囲に含まれる、全ての座標について、
前記全ての座標が対応する点の集合に関する統計量を計算する、ステップと、
を有する、ことを特徴とする情報処理方法。

[請求項12] (d) 前記 (a) のステップによって特定された区間が複数である場合に、前記 (b) のステップによって計算された区間毎の前記点の集合に関する統計量を、更に集計し、集計によって得られた統計量を、前記クエリ領域に含まれる点の集合に関する全体的な統計量として、出力する、ステップを更に有する、
請求項 1 1 に記載の情報処理方法。

[請求項13] 前記データ構造が、前記 (a) のステップによる前記区間の特定に用いられる第 1 のデータ構造と、前記 (c) のステップによる前記統計量の計算に用いられる第 2 のデータ構造とを有している、
請求項 1 1 または 1 2 に記載の情報処理方法。

[請求項14] 前記第 1 のデータ構造が、
前記多次元空間に設定された複数のカバー領域のいずれかと、前記点の列上において当該カバー領域に含まれる点が出現する区間とに関連付けられている、ノードを有する木構造によって表現されており、
前記 (a) のステップにおいて、

前記ノードのうち、

関連付けられた前記カバー領域に存在する点における、前記1つの次元を除いた残りの各次元の座標がクエリ領域に含まれている、ノードを特定し、

特定した1又は2以上のノードが関連付けられている区間を前記区間として特定する、

請求項13に記載の情報処理方法。

[請求項15] 前記点の列は、

前記ノードに関連付けられたカバー領域それぞれに存在する点が、ひとつつながりで連続して出現するように、前記点の集合に含まれる点を一列に並べて得られている、

請求項14に記載の情報処理方法。

[請求項16] 前記(c)のステップにおいて、前記第2のデータ構造を用いて、前記座標列から得られる複数の部分列のうち、入力された前記範囲に含まれる座標のみが出現する部分列を特定し、そして、特定した部分列上の区間であって、前記座標列において入力された前記区間に出現する座標が出現している、第二の区間を特定し、
更に、特定した前記第二の区間に出現する座標が対応する点の集合に関する統計量を計算する、

請求項13に記載の情報処理方法。

[請求項17] 前記部分列が、座標のビット表現が同じ接頭辞で始まる座標を、座標同士的位置関係を保ったまま抽出することで得られるものであり、

前記第2のデータ構造が、

前記部分列に関連付けられた複数のノードを有し、

前記複数のノードそれぞれは、前記部分列において出現する各座標のビット表現における、一つ以上の特定の桁のビットを取り出し、取り出した前記ビットを前記部分列と同じ順序で並べることによって得られる、ビットの列を用いて表現され、

前記（c）のステップにおいて、前記複数ノードそれぞれを表現するビットの列を用いて、前記第二の区間を特定する、
請求項 16 に記載の情報処理方法。

[請求項18] 前記（c）のステップにおいて、前記全ての座標が対応する点の集合に関する統計量として、前記全ての座標が対応する点の個数を計算する、
請求項 11 ～ 17 のいずれかに記載の情報処理方法。

[請求項19] 前記（c）のステップにおいて、前記全ての座標が対応する点の集合に関する統計量として、前記全ての座標が対応する点それぞれの各次元の座標を計算する、
請求項 11 ～ 17 に記載の情報処理方法。

[請求項20] 多次元空間上の点の集合を表現するデータ構造を処理対象とする情報処理をコンピュータによって行なうためのプログラムを記録したコンピュータ読み取り可能な記録媒体であって、
前記コンピュータに、

（a）クエリ領域として、特定の多次元の領域が指定された場合に、前記点の集合を一行に並べて得られた点の列上にあり、且つ、前記多次元空間を構成する全次元のうち1つの次元を除いた残りの各次元の座標がクエリ領域に含まれている点のみによって構成されている、区間を特定する、ステップと、

（b）前記（a）のステップで特定された区間について、当該区間に出現する点が前記クエリ領域に含まれるための条件として、除かれた前記1つの次元における座標の値の範囲を特定する、ステップと、

（c）前記（a）のステップで特定された区間、及び前記（b）のステップで特定された前記座標の値の範囲を入力として、
前記点の列の並び順と同じ順序で、前記点の集合の各点における、除かれた前記1つの次元での座標を取り出すことで得られる、座標列に関して、当該座標列において入力された前記区間に出現し、且つ、値

が入力された前記範囲に含まれる、全ての座標について、
前記全ての座標が対応する点の集合に関する統計量を計算する、ステップと、
を実行させる命令を含む、プログラムを記録しているコンピュータ読み取り可能な記録媒体。

[請求項21] 前記プログラムが、前記コンピュータに、
(d) 前記(a)のステップによって特定された区間が複数である場合に、前記(b)のステップによって計算された区間毎の前記点の集合に関する統計量を、更に集計し、集計によって得られた統計量を、前記クエリ領域に含まれる点の集合に関する全体的な統計量として、出力する、ステップを実行させる命令を更に含む、
請求項20に記載のコンピュータ読み取り可能な記録媒体。

[請求項22] 前記データ構造が、前記(a)のステップによる前記区間の特定に用いられる第1のデータ構造と、前記(c)のステップによる前記統計量の計算に用いられる第2のデータ構造とを有している、
請求項20または21に記載のコンピュータ読み取り可能な記録媒体。

[請求項23] 前記第1のデータ構造が、
前記多次元空間に設定された複数のカバー領域のいずれかと、前記点の列上において当該カバー領域に含まれる点が出現する区間とに関連付けられている、ノードを有する木構造によって表現されており、
前記(a)のステップにおいて、
前記ノードのうち、
関連付けられた前記カバー領域に存在する点における、前記1つの次元を除いた残りの各次元の座標がクエリ領域に含まれている、ノードを特定し、
特定した1又は2以上のノードが関連付けられている区間を前記区間として特定する、

請求項 2 2 に記載のコンピュータ読み取り可能な記録媒体。

[請求項24]

前記点の列は、

前記ノードに関連付けられたカバー領域それぞれに存在する点が、ひとつつながりで連続して出現するように、前記点の集合に含まれる点を一列に並べて得られている、

請求項 2 3 に記載のコンピュータ読み取り可能な記録媒体。

[請求項25]

前記 (c) のステップにおいて、前記第 2 のデータ構造を用いて、前記座標列から得られる複数の部分列のうち、入力された前記範囲に含まれる座標のみが出現する部分列を特定し、そして、特定した部分列上の区間であって、前記座標列において入力された前記区間に出現する座標が出現している、第二の区間を特定し、
更に、特定した前記第二の区間に出現する座標が対応する点の集合に関する統計量を計算する、

請求項 2 2 に記載のコンピュータ読み取り可能な記録媒体。

[請求項26]

前記部分列が、座標のビット表現が同じ接頭辞で始まる座標を、座標同士の位置関係を保ったまま抽出することで得られるものであり、

前記第 2 のデータ構造が、

前記部分列に関連付けられた複数のノードを有し、

前記複数のノードそれぞれは、前記部分列において出現する各座標のビット表現における、一つ以上の特定の桁のビットを取り出し、取り出した前記ビットを前記部分列と同じ順序で並べることによって得られる、ビットの列を用いて表現され、

前記 (c) のステップにおいて、前記複数ノードそれぞれを表現するビットの列を用いて、前記第二の区間を特定する、

請求項 2 5 に記載のコンピュータ読み取り可能な記録媒体。

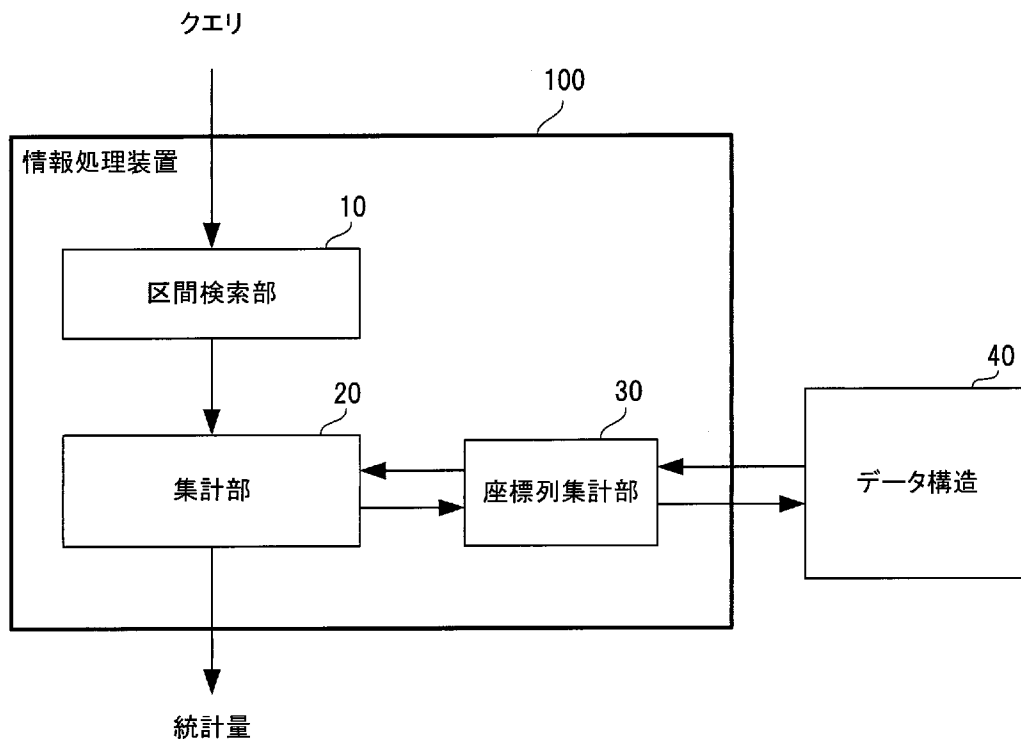
[請求項27]

前記 (c) のステップにおいて、前記全ての座標が対応する点の集合に関する統計量として、前記全ての座標が対応する点の個数を計算する、

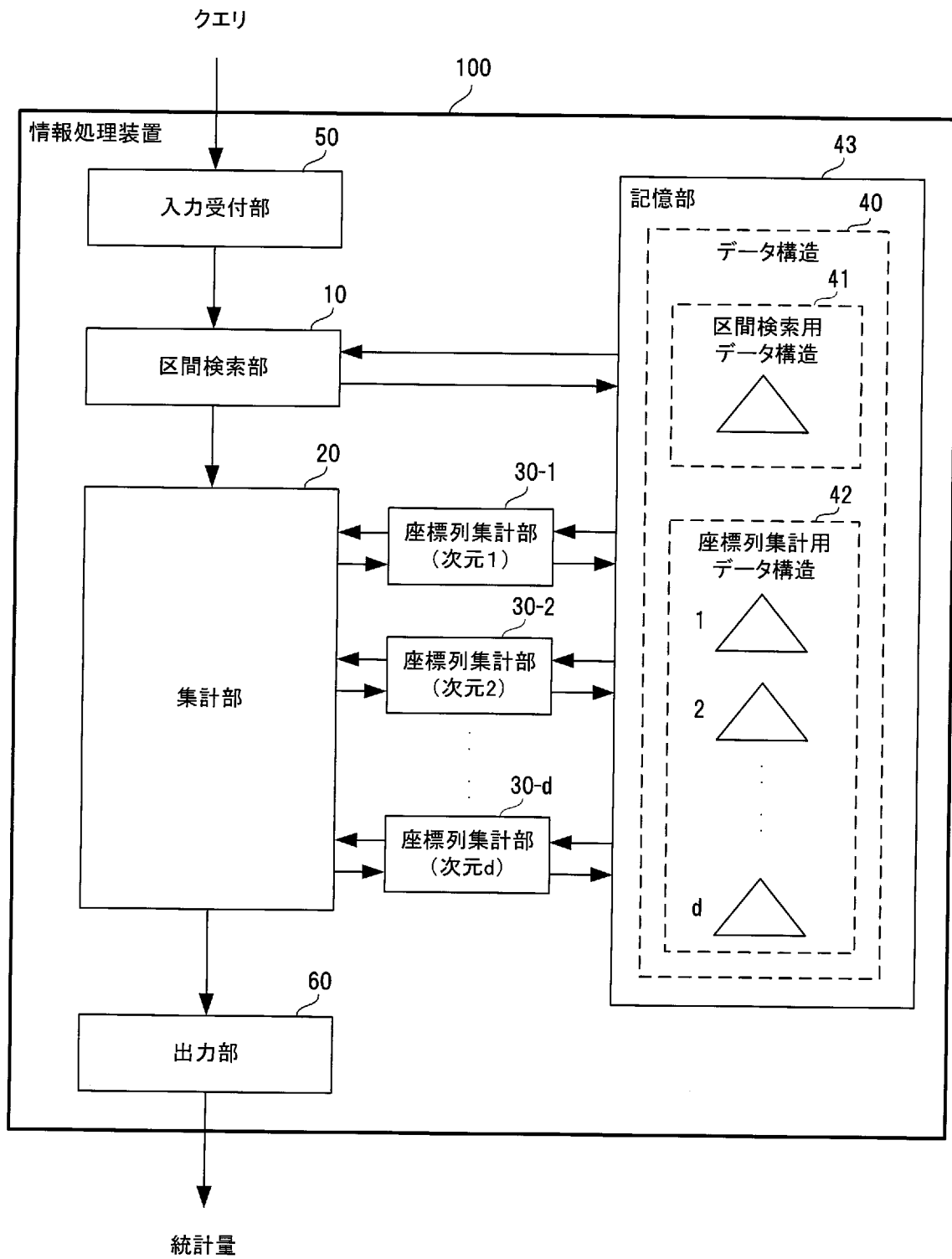
請求項 20～26 のいずれかに記載のコンピュータ読み取り可能な記録媒体。

[請求項28] 前記（c）のステップにおいて、前記全ての座標が対応する点の集合に関する統計量として、前記全ての座標が対応する点それぞれの各次元の座標を計算する、
請求項 20～26 に記載のコンピュータ読み取り可能な記録媒体。

[図1]



[図2]



[図3]

2次元平面(空間)における点

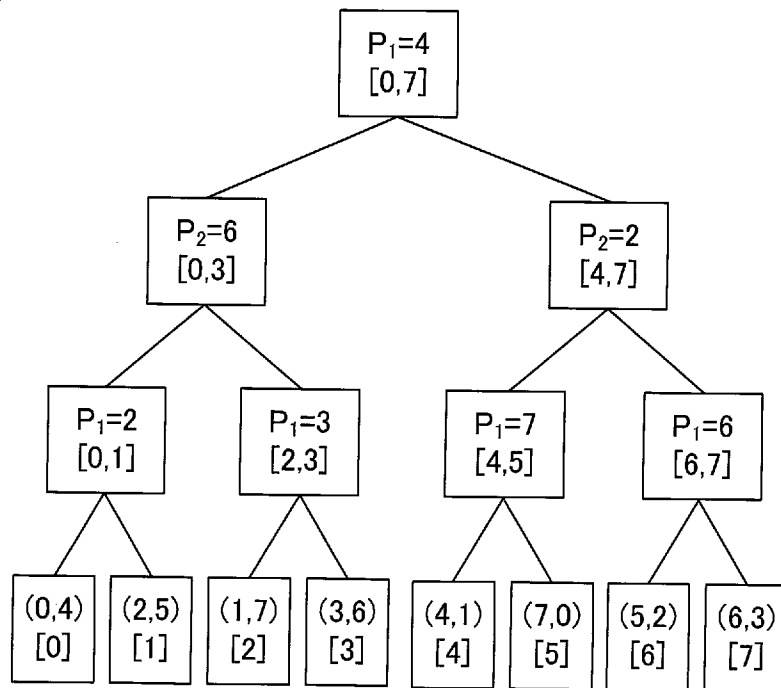
Dimension 2

Dimension 1

	0 "000"	1 "001"	2 "010"	3 "011"	4 "100"	5 "101"	6 "110"	7 "111"
0 "000"					①			
1 "001"								②
2 "010"						①		
3 "011"							③	
4 "100"		④						
5 "101"			⑥					
6 "110"				⑦				
7 "111"	⑤							

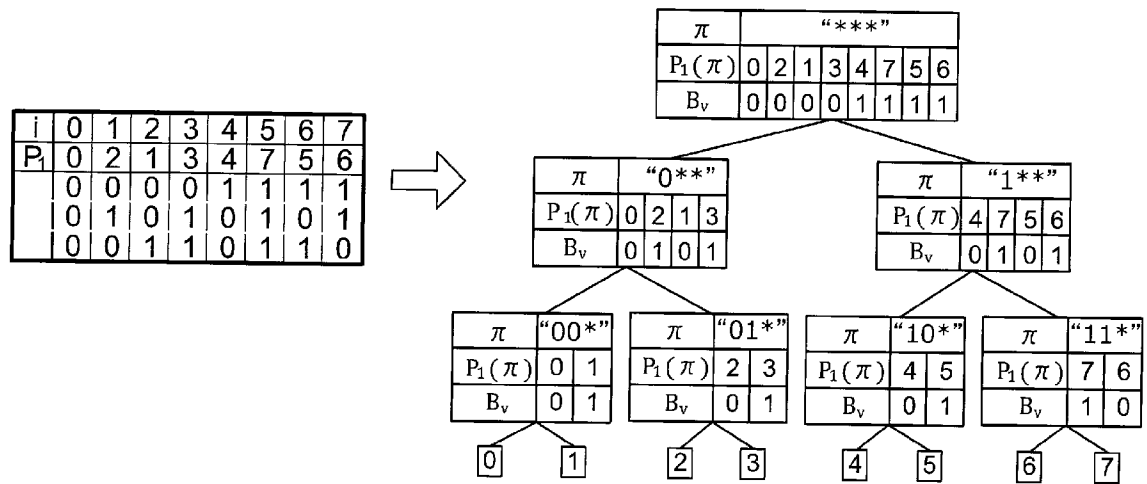
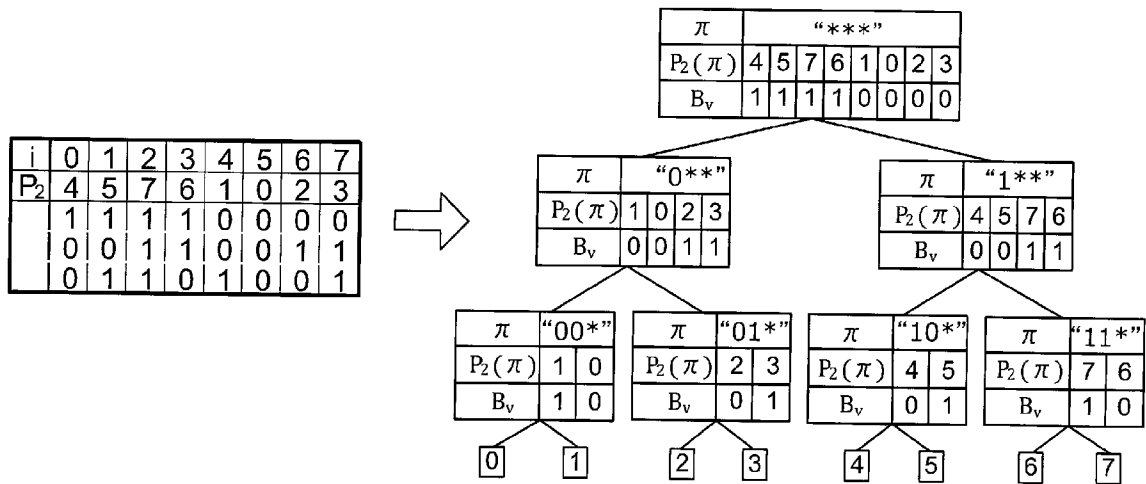
[図4]

(a) kd木

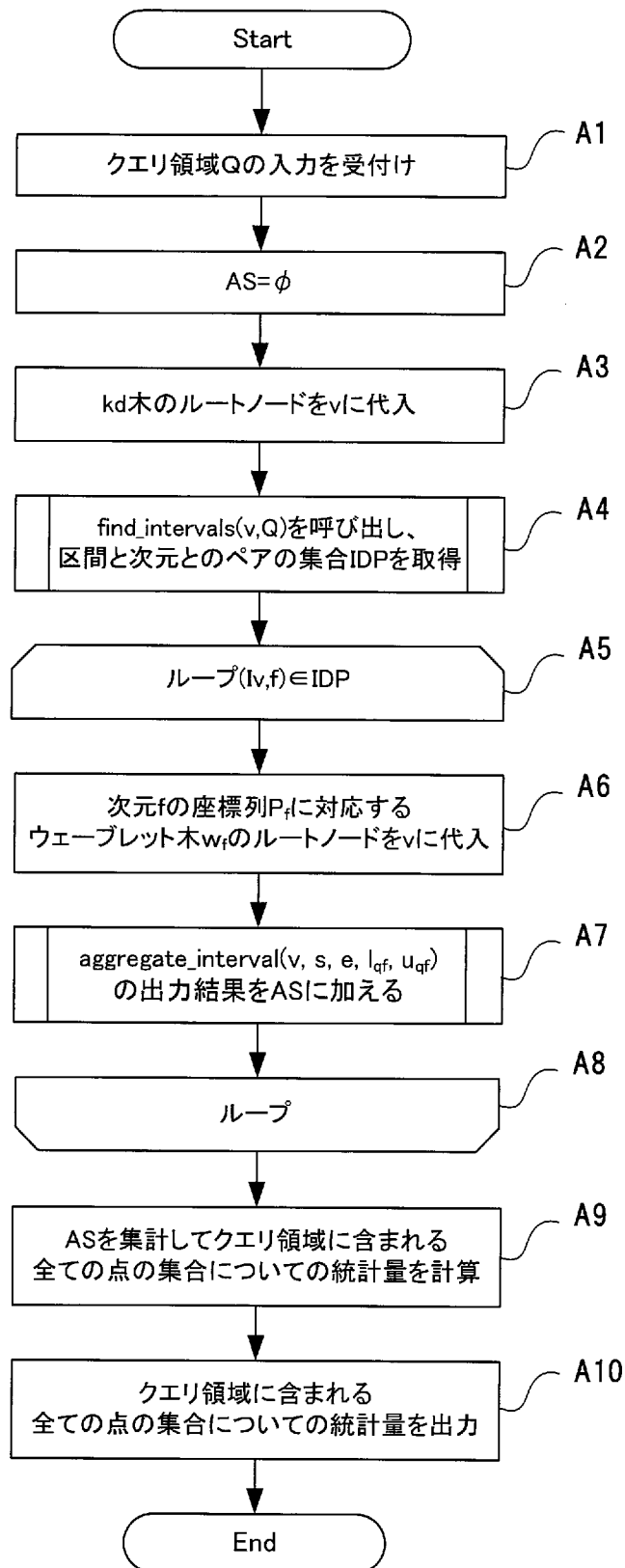
(b) 点の列P(座標列 P_1 および座標列 P_2)

i	0	1	2	3	4	5	6	7
P_1	0	2	1	3	4	7	5	6
P_2	4	5	7	6	1	0	2	3

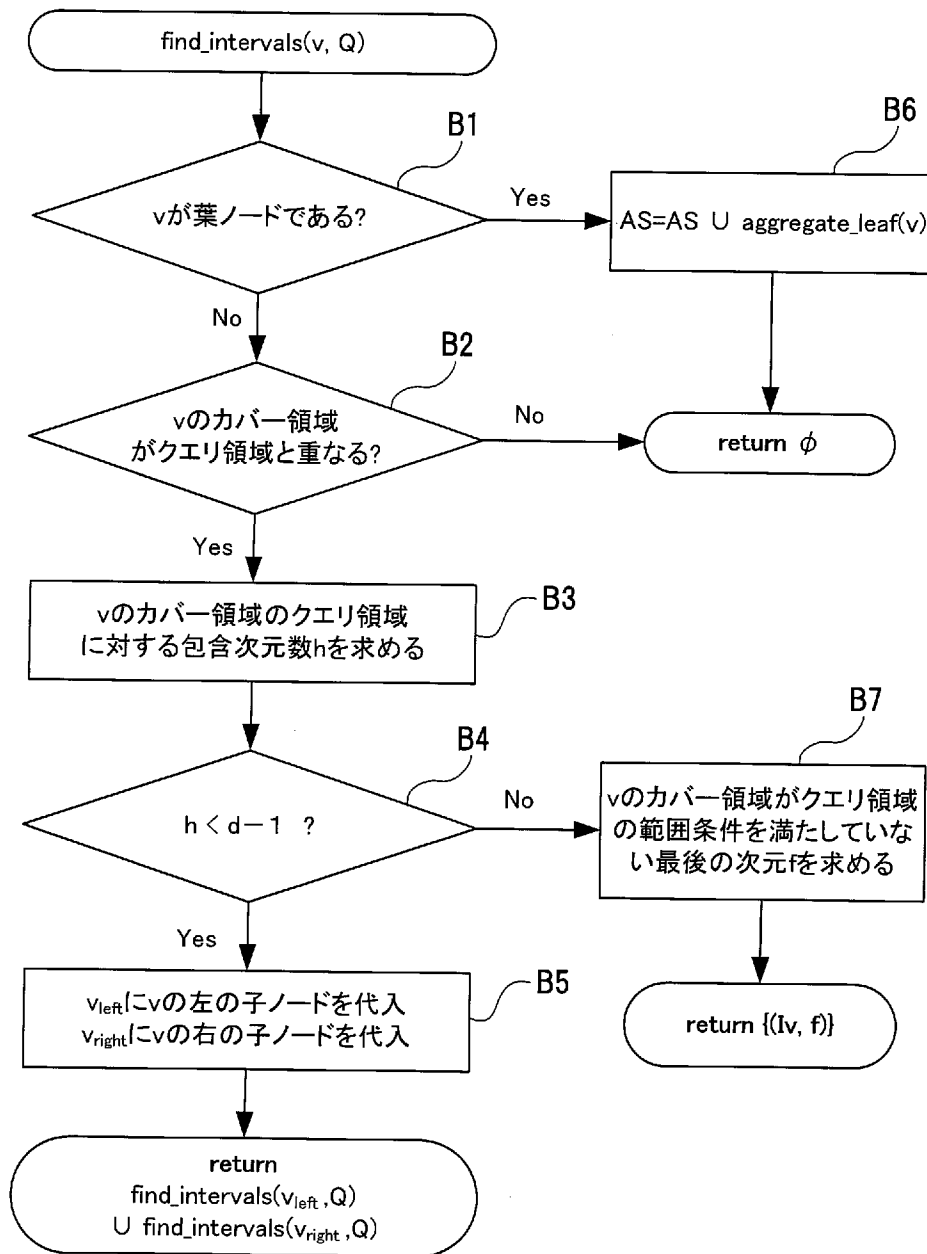
[図5]

(a) 座標列 P_1 のウェーブレット木 w_1 (b) 座標列 P_2 のウェーブレット木 w_2 

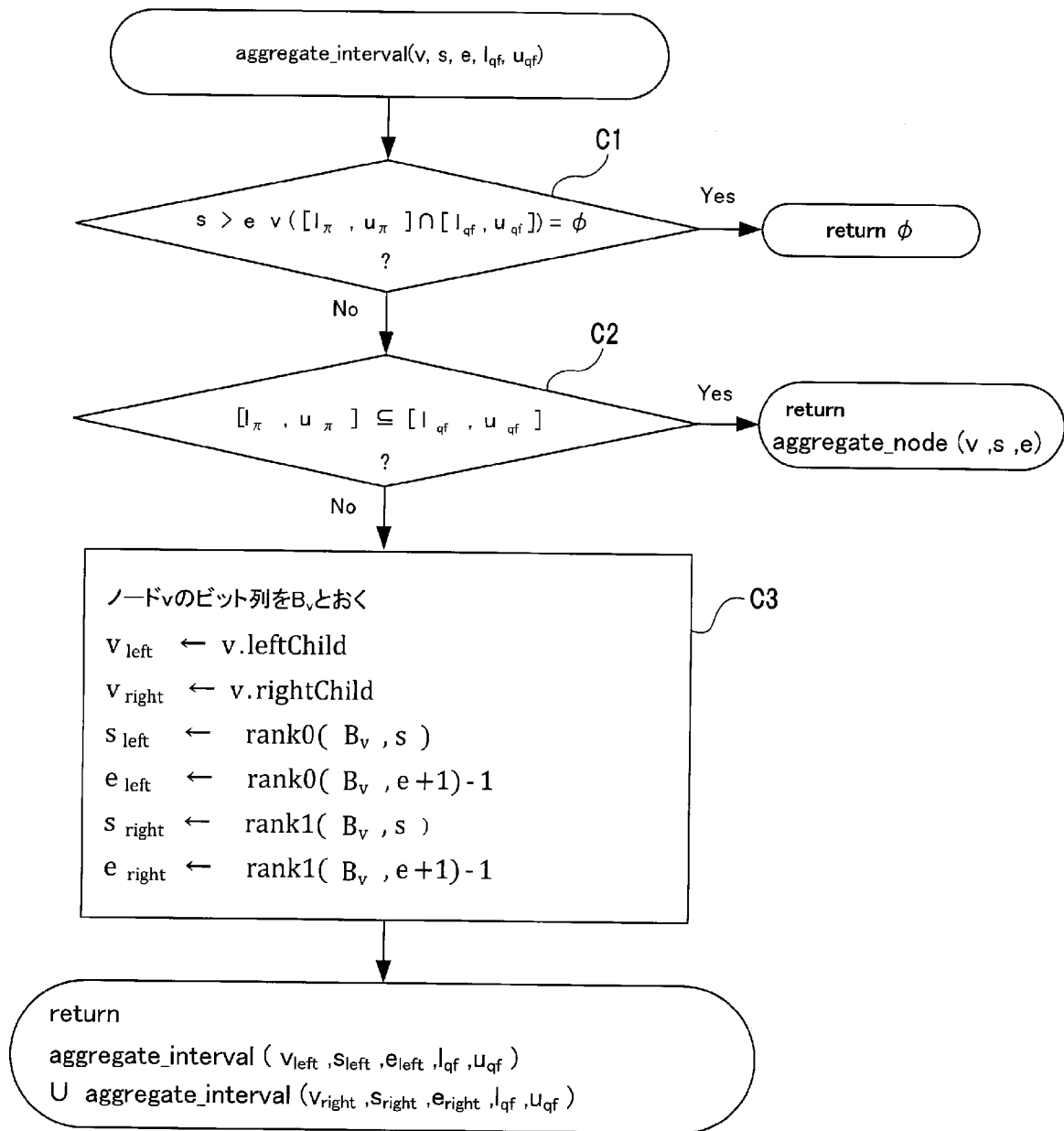
[図6]



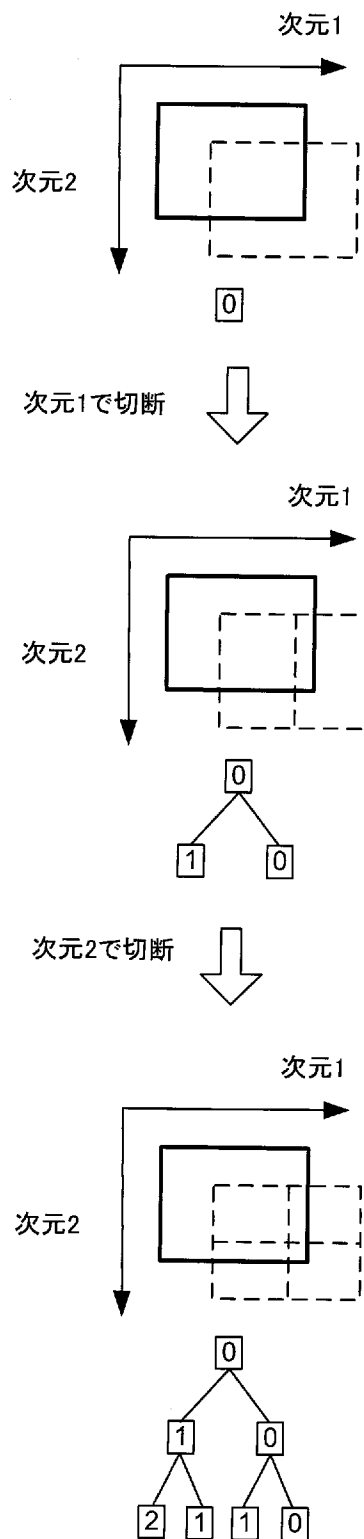
[図7]



[図8]



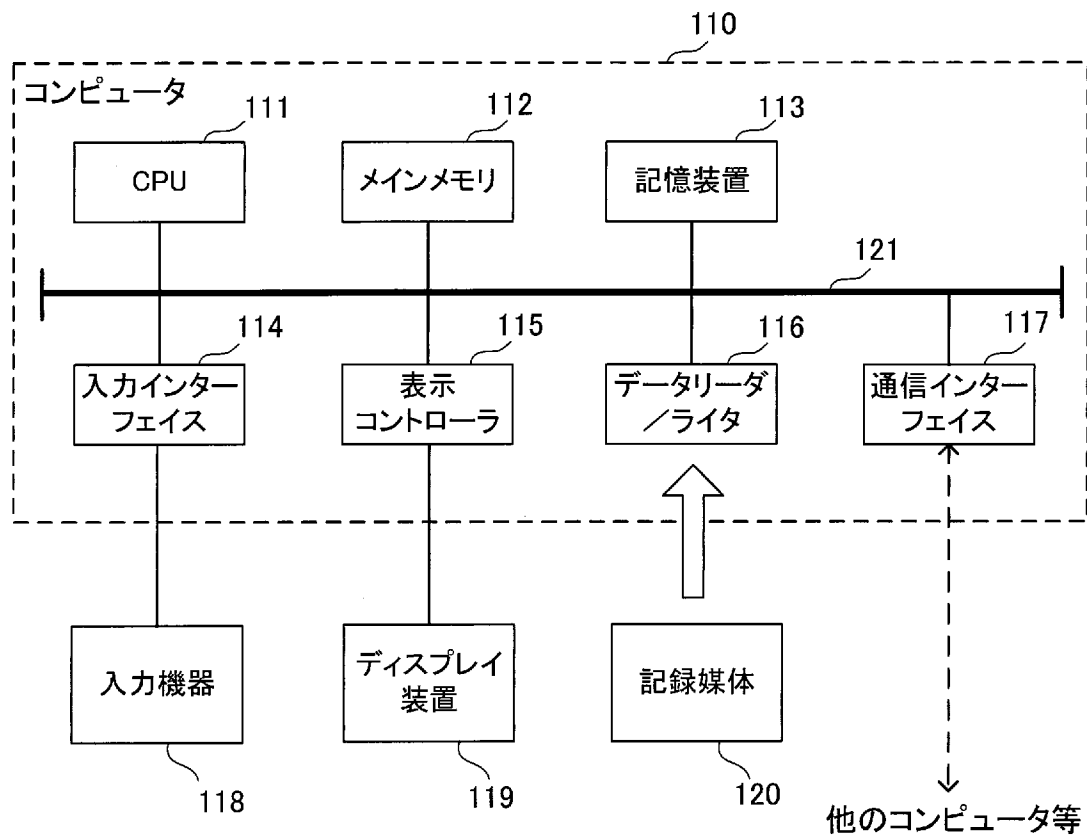
[図9]



[図10]

	矩形範囲カウント		矩形範囲レポート	
	d=2	d>2	d=2	d>2
kd木	$O(\sqrt{n})$	$O(n^{(d-1)/d})$	$O(\sqrt{n} + F)$	$O(n^{(d-1)/d} + F)$
ウェーブレット木	$O(\log n)$	適用できない	$O(\log n + F \log n)$	適用できない
本発明	$O(\log^2 n)$	$O(n^{(d-2)/d} \log n)$	$O(\log^2 n + F \log n)$	$O(n^{(d-2)/d} \log n + F \log n)$

[図11]



INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2015/079476

A. CLASSIFICATION OF SUBJECT MATTER

G06F17/30(2006.01) i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F17/30

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Jitsuyo Shinan Koho	1922-1996	Jitsuyo Shinan Toroku Koho	1996-2016
Kokai Jitsuyo Shinan Koho	1971-2016	Toroku Jitsuyo Shinan Koho	1994-2016

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	JP 10-293863 A (Power Reactor and Nuclear Fuel Development Corp.), 04 November 1998 (04.11.1998), entire text; all drawings & US 6072495 A & DE 19817584 A1 & CA 2235089 A1	1-28
A	JP 2014-002519 A (Hitachi, Ltd.), 09 January 2014 (09.01.2014), entire text; all drawings & US 9098591 B2 & US 2013/0339371 A1 & CN 103514243 A	1-28
A	JP 9-265479 A (Hitachi, Ltd.), 07 October 1997 (07.10.1997), entire text; all drawings & US 5864857 A	1-28

☒ Further documents are listed in the continuation of Box C.

☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search
06 January 2016 (06.01.16)

Date of mailing of the international search report
19 January 2016 (19.01.16)

Name and mailing address of the ISA/
Japan Patent Office
3-4-3, Kasumigaseki, Chiyoda-ku,
Tokyo 100-8915, Japan

Authorized officer

Telephone No.

INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2015/079476

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	Yoshitomo YAGINUMA et al., "One Proposal of a N-Dimensional Multi-Layer Data Structure with Unified Spatial and Attributive Management Capability", The Transactions of the Institute of Electronics, Information and Communication Engineers, 25 February 1992 (25.02.1992), vol. J75-D-I, no.2, pages 88 to 97	1-28

A. 発明の属する分野の分類（国際特許分類（IPC））

Int.Cl. G06F17/30(2006.01)i

B. 調査を行った分野

調査を行った最小限資料（国際特許分類（IPC））

Int.Cl. G06F17/30

最小限資料以外の資料で調査を行った分野に含まれるもの

日本国実用新案公報	1922-1996年
日本国公開実用新案公報	1971-2016年
日本国実用新案登録公報	1996-2016年
日本国登録実用新案公報	1994-2016年

国際調査で使用した電子データベース（データベースの名称、調査に使用した用語）

C. 関連すると認められる文献

引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求項の番号
A	JP 10-293863 A（動力炉・核燃料開発事業団） 1998.11.04, 全文, 全図 & US 6072495 A & DE 19817584 A1 & CA 2235089 A1	1-28
A	JP 2014-002519 A（株式会社日立製作所） 2014.01.09, 全文, 全図 & US 9098591 B2 & US 2013/0339371 A1 & CN 103514243 A	1-28

☑ C欄の続きにも文献が列挙されている。

☐ パテントファミリーに関する別紙を参照。

* 引用文献のカテゴリー

「A」特に関連のある文献ではなく、一般的技術水準を示すもの
「E」国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの
「L」優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献（理由を付す）
「O」口頭による開示、使用、展示等に言及する文献
「P」国際出願日前で、かつ優先権の主張の基礎となる出願

の日の後に公表された文献

「T」国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの
「X」特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの
「Y」特に関連のある文献であって、当該文献と他の1以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの
「&」同一パテントファミリー文献

国際調査を完了した日

06.01.2016

国際調査報告の発送日

19.01.2016

国際調査機関の名称及びあて先

日本国特許庁（ISA/J P）

郵便番号100-8915

東京都千代田区霞が関三丁目4番3号

特許庁審査官（権限のある職員）

齊藤 貴孝

5M

4774

電話番号 03-3581-1101 内線 3599

C (続き) . 関連すると認められる文献		
引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求項の番号
A	JP 9-265479 A (株式会社日立製作所) 1997. 10. 07, 全文, 全図 & US 5864857 A	1-28
A	柳沼 良知、外 1 名, 位置情報と属性情報の複合管理を可能とする データ構造の一提案, 電子情報通信学会論文誌, 1992. 02. 25, 第 J 7 5 - D - I 巻, 第 2 号, p. 8 8 - 9 7	1-28