

相対的位置情報を用いた 3 次元グリッドによる位置検索法

佐藤 玲奈[†] 増永 良文[‡]

[†]お茶の水女子大学人間文化研究科数理・情報科学専攻 〒112-8610 東京都文京区大塚 2-1-1

[‡]お茶の水女子大学理学部情報科学科 〒112-8610 東京都文京区大塚 2-1-1

E-mail: [†]reina@db.is.ocha.ac.jp, [‡]masunaga@is.ocha.ac.jp

あらまし 本稿では、<前に何々が見える>というような間接的表現である“相対的な位置情報”を用い 3 次元位置検索を考案した。相対的位置情報から位置を推定することは空間インデックス作成(R-Tree, 四分木), 空間演算など通常であれば多くの計算を必要とする。そこで、新たに提案する連続空間を近似した 3 次元グリッドを用いることで、道路と建物を簡素モデル化し、建物を構成する各グリッドに対し、視点となるグリッドからの可視・不可視判定の事前計算を可能とし、事前計算結果と相対的位置情報を用いた簡便な計算による位置推定法を提案する。

キーワード 相対的位置情報, 地図, 位置検索法, 3 次元グリッド

A Location Identification Method using Relative Position Information and 3-Dimensional Grid

Reina SATO[†] and Yoshifumi MASUNAGA[‡]

[†] Graduate School of Humanities and Sciences, Ochanomizu University

2-1-1 Otsuka, Bunkyo-ku, Tokyo, 112-8610 Japan

[‡] Department of Information Science, Faculty of Science Ochanomizu University

2-1-1 Otsuka, Bunkyo-ku, Tokyo, 112-8610 Japan

E-mail: [†]reina@db.is.ocha.ac.jp, [‡]masunaga@is.ocha.ac.jp

Abstract This paper proposes a 3-dimension location identification method that uses "relative location information" like "I can see a department store ahead.". Presuming the position from relative position information using spatial indexes (ex. R-Tree, QuadTree, etc.) or spatial operations usually consume much of computing resources. We propose simple position presumption method that uses pre-computed result and relative position information. The pre-computed result mainly consist of both simplification and visibility, the former is making a real world into a 3 dimensional grid virtual world structured with simplified building and road shapes, and the latter is a visible grid detection from each viewpoint on grid.

Keyword Relative Location Information, Map, Location Identification Method, 3-dimensional Grid

1. はじめに

近年、モバイルコンピューティング技術やコンピュータの高性能化、小型化が進み、コンピュータを持ち歩き様々な情報を享受する環境が普及してきた。位置情報を GPS にて取得できる携帯端末も一般的になり、位置情報は企業活動や市民生活に不可欠な存在となりつつある。しかし、ヒトとヒトの間の意思疎通はこれまで曖昧表現で位置情報を伝達しており、自位置を伝える際に緯度経度や住所などの直接的表現ではなく、「前に何々が見える、右に何々が見える」のような間接的表現を用いている。このような建物名と方位を組み合わせた表現を相対的位置情報と定義する。相対的位置情報を用いてヒトの位置が推定可能になると日常の様々なシーンで活用できうる。

本稿では、相対的位置情報を用いた 3 次元位置検索法を考えた。相対的位置情報から位置を推定することは空間インデックス作成(R-Tree, 四分木), 空間演算など通常であれば多くの計算を必要とするが、新たに提案する道路や建物オブジェクトの 3 次元グリッドを用いることで簡便な計算による位置推定を可能とする。また 3 次元グリッドを活用することは、2 次元グリッドを用いた位置推定よりも現実世界の投影能力が高い。例えば「前に X デパートが見えて、後に Y コンビニエンスストアが見える・・・」という情報は<他の建物の影にならずに X デパートが見える>という意味であり、3 次元グリッドを用いることでより正確性をもつ位置推定を実現する。ある地点から建物オブジェクトが「見える, 見えない」は通常多くの計算を要するが、

3次元グリッドを用いて道路・建物オブジェクトを簡素モデル化することで事前計算を可能とし、その結果を用い瞬時的な位置推定を行う。3次元グリッドを用いた簡便な位置推定は現実世界の投影であり、非力なコンピューティング環境でも実用性が高いため、モバイル携帯端末などでも応用可能であろう。

以下、2章では本稿の位置検索システムの概略について、3章ではオブジェクトグリッドの可視・不可視判定法について、4章では相対的位置情報を用いたユーザ位置検索法について、5章では実装について述べ、6章でまとめと今後の課題について述べる。

2. 位置検索システムの概略

2.1. 相対的位置情報

ヒトとヒトとの間の意思疎通は曖昧表現で位置情報を伝達しており、自位置を相手に伝達するには緯度経度や、住所などの直接的表現ではなく、ヒトがおかれている外界の中で視認した周囲の建物名と、自分と建物オブジェクトとの空間的関係を用いている。本来、空間における2つのオブジェクト間には空間的関係の3大要素である距離関係、方向関係、位相関係が考えられる[1][2]。距離関係は、ユーザの認識誤差が出ることや、位相関係は現実世界では disjoint か meet かという判別のみになることから、本稿では、ユーザからの建物オブジェクトの方向関係に着目する。方向関係には「東、西、南、北」概念である絶対的方向関係と、「前、後、左、右」+「上、下」概念である相対的方向関係がある。我々は位置情報を把握する際、建物オブジェクトが自分の、「前、後、左、右」+「上、下」いずれの方向に見えるかという関係を用いている場合が多いことから、ユーザの向きを基準とし、ユーザから視認できる建物オブジェクトの方向を考える、ユーザ基準の相対的方向関係を導入する。それにより例えば「前方斜め上にXデパートが見える」というような表現が可能となる。

2.2. システム概要

相対的位置情報を用いた位置検索システム概要を図1に示す。地図DBより3次元グリッドから道路、建物オブジェクトを構成する内部の3次元グリッドを抽出する(以下、オブジェクトグリッドとする)。抽出したオブジェクトグリッドを用い可視・不可視判定する(以下、可視オブジェクトグリッドとする)ことで、地図データを事前計算により索引付けする。ユーザ入力(KeyもしくはVoice)による相対的位置情報と索引を用い瞬時的な位置検索を行い、結果を地図表示しユーザ位置推定を行う。

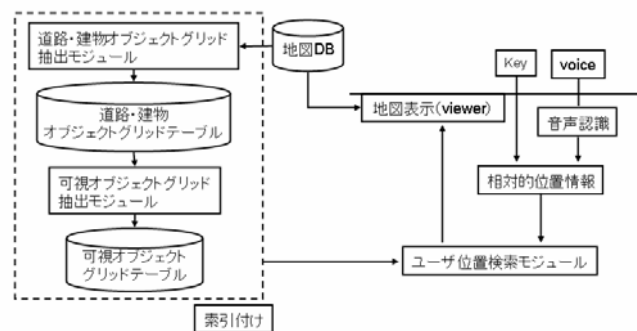


図1: システム概要

Fig1.A System Overview

3. オブジェクトグリッド可視・不可視判定法

本章では、3次元グリッドを用い道路、建物オブジェクトを簡素モデル化することで現実世界を投影し、可視・不可視判定を事前計算することで、地図データを索引付けする方法について述べる。

3.1. 3次元グリッド近似法

相対的位置情報を用いた位置検索の計算方法として、例えば3つの建物オブジェクトが視認された場合、2つの建物オブジェクトの重心を結ぶ線分への他の建物オブジェクトの重心からの垂線で、位置を求める方法があるが、この場合建物オブジェクトの形状が考慮されず、建物オブジェクトの重心だけの方向関係に制約されてしまう。そのため建物オブジェクトの多様な形状を考慮した計算を行う場合、本来、空間は連続空間であるためユーザの位置は「点」では求まらず、ある「範囲」に決定される[3,4,5]。またユーザが指定する相対的方向関係は時として大雑把であり、“前”に見えるといっても正面ではないこともあるため、方向指定に曖昧性を許容できるモデル化が望ましい。そこで、本稿では連続空間を3次元グリッド近似するモデル化を考えた。それによりユーザの尤もらしい位置を検索でき、計算コストの削減にも繋がる。

3次元グリッド生成法は、一定の経線、緯線で地域を網の目状に区画する方法のメッシュ概念の一つである。経度幅1/8度、緯度幅2/3度の約10×10kmである2次元メッシュ(国土地理院)ごとに、x,y方向に各々N等分でメッシュ化し、さらにz方向にメッシュ化することで3次元メッシュを作成し、連続空間を生成された6面体の3次元グリッドとする。各3次元グリッドは左下、右上の緯度経度データで表現可能であるが、膨大データが対象であるためより計算量を削減するために生成した各3次元グリッドをID(x,y,z)で表す。ここにx,y,zは分割された2次元メッシュの左下を原点とし非負整数で表すとする。各2次元メッシュの経度、

緯度幅が各々1/8度, 2/3/8度であることから, 各3次元グリッドの経度, 緯度幅は1/8/N度, 2/3/8/N度となり, 2次メッシュコード, ID(x, y, z), 分割数より3次元グリッドの左下頂点の緯度経度は, 2次メッシュの左下の経度に幅(1/8/N)*x, 緯度に幅(2/3/8/N)*yを加えることで計算可能である. 逆も同様に計算可能である.

3.2. オブジェクトグリッド簡素モデル化

上記方法で生成した3次元グリッドと地図データの道路, 建物データとの衝突判定により道路, 建物オブジェクトを構成するオブジェクトグリッドを抽出する. 建物オブジェクトの場合, 図2のように地図データの建物オブジェクトのレイヤと3次元グリッドを衝突判定し接し, かつ含まないグリッドを側面とし, 接するグリッドを天井とし建物を構成する建物オブジェクトグリッドとする. 道路の場合, 道路のレイヤと衝突判定し接するグリッドで, 標高を考慮しz方向が最も高いグリッドを道路オブジェクトグリッドとする. 抽出されたオブジェクトグリッドをオブジェクトグリッドテーブルに格納する. テーブルのカラムは GeometryID, Layercode, 2次メッシュコード, ID(x, y, z)とする. ここで GeometryID は地図データで各道路や建物オブジェクトに一意に与えられている ID, Layercode は各レイヤに与えられている番号(例えば公共建物の番号: 497)である. この簡素モデル化は, ユーザが指定する方向関係は時として大雑把であり, “前”に見えるといっても誤差を含むこともあるため, 方向指定に曖昧性を許容するモデル化となっている.

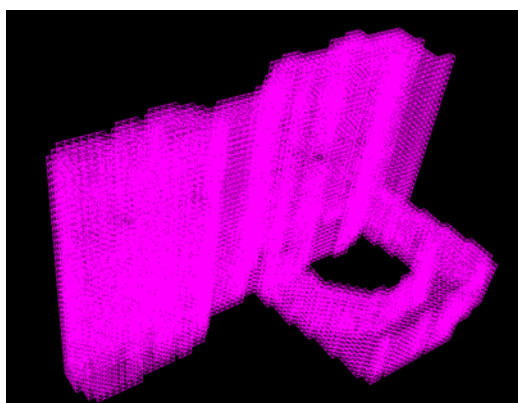


図2: オブジェクトグリッド簡素モデル化
Fig2. Object Grid Simplification Model

ここで, R, B, G^{RUB} を次のように定義する.

- 抽出した道路オブジェクトグリッド集合

$$R = \{r_i | i = 1, \dots, n\}$$

- 抽出した建物オブジェクトグリッド集合

$$B = \{b_i | i = 1, \dots, m\}$$

- 道路と建物オブジェクトグリッド集合

$$G = R \cup B$$

$$= \{r_i | i = 1, \dots, n\} \cup \{b_i | i = 1, \dots, m\}$$

3.3. 可視判定対象の視野範囲

抽出したオブジェクトグリッドを用い, 視点となるオブジェクトグリッド G から可視オブジェクトグリッドを自動抽出する. 視点となるオブジェクトグリッド G からのオブジェクトグリッド G の可視性をすべて事前計算しテーブルに格納すると膨大な計算量となる. そこで, 視点となるオブジェクトグリッド G から x-y 平面にて任意の範囲(例えば2次メッシュを2000分割した際のグリッド数20以内)を視野とし, 範囲内にある建物オブジェクトグリッド B を可視判定計算対象とする.

3.4. 可視オブジェクトグリッド抽出アルゴリズム

次に視点となるオブジェクトグリッド G から可視オブジェクトグリッドを自動抽出するアルゴリズムを述べる(図3). 視点が道路オブジェクトグリッド R とは, ユーザが道路上にいる場合を想定し(身長はグリッドの一辺の長さに比べると小さいと仮定), それが建物オブジェクトグリッド B とは, ユーザが建物の窓から対象を見ている場合である.

視点となる各オブジェクトグリッド G に対して以下を行う.

1. 視点となるオブジェクトグリッド G から x-y 平面にて任意の範囲内にある建物オブジェクトグリッド B を抽出する.
 2. 視点となるオブジェクトグリッド G の重心から各建物オブジェクトグリッド B へ線分を引く.
 3. x-y 平面にて, 線分を含む MBR 内(視点となるオブジェクトグリッド G と可視判定対象の建物オブジェクトグリッド B は除く)にて, 線分と他の建物オブジェクトグリッド B との衝突判定する.
- 衝突しなかった場合, 可視判定対象の建物オブジェクトグリッド B の z 方向分を含めたすべてを可視オブジェクトグリッドとする.
 - 衝突した場合, y-z 平面にて可視判定対象の建物オブジェクトグリッド B へ線分を引き, 同様の衝突判定を行い, 他の建物オブジェクトグリッド B と

衝突しなかった可視判定対象の建物オブジェクトグリッド B を、可視オブジェクトグリッドとする。

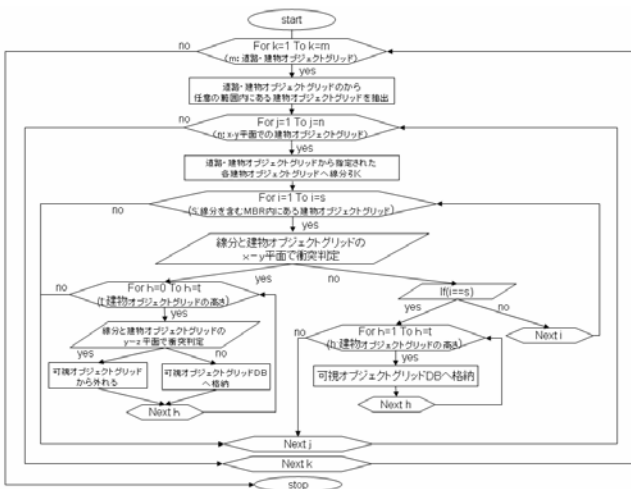


図 3：可視オブジェクトグリッド抽出アルゴリズム
Fig3. Visible Object Grid Extraction Algorithm

例えば図 4 で視点となる道路オブジェクトグリッド R から建物オブジェクトグリッド B に引いた線分で、 x - y 平面では建物オブジェクトグリッド B に線分が衝突するため、 y - z 平面での衝突判定を行い、上二つの建物オブジェクトグリッド B が可視オブジェクトグリッドとなる。

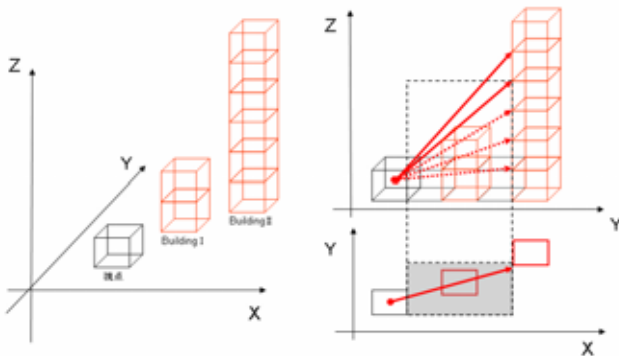


図 4：可視オブジェクトグリッド判別例
Fig4. Example of Distinguishing Visible Object Grid

抽出した可視オブジェクトグリッドをテーブルに格納する。視点となるオブジェクトグリッド [道路, 建物] の 2 次メッシュ番号と ID (x, y, z), それに対する可視オブジェクトグリッドの 2 次メッシュコードと ID (x, y, z), GeometryID をカラムとする。

ここで、 $Vis, VisG$ を次のように定義する。

- ・ 視点 1 点からの可視オブジェクトグリッド集合

$$Vis = \{ vis_i \mid i = 1, \dots, l \}$$

- ・ オブジェクトグリッド G からの可視オブジェクトグリッド集合

$$VisG = \{ visg_G^i \mid i = 1, \dots, k \}$$

3.5. 地図データの索引付けの有効性検証実験

以上の方法により、視点となる各オブジェクトグリッド G からの可視オブジェクトグリッド $VisG$ の自動抽出を行いテーブルに格納することで、地図データを事前計算により索引付けする。空間演算は通常の場合、その都度必要に応じて計算を行うため多くの計算が必要とされるが、本稿では、3 次元グリッドを利用し地図データを索引付けすることで、索引から取得可能な、視認した建物オブジェクトがすべて見えるオブジェクトグリッド G と相対的方向関係に基づき、建物オブジェクトの可視部分のみを対象とした瞬時的なユーザ位置推定が可能である。

実際に、索引付けの有効性を定量的に検証する実験として、ある視点から可視オブジェクトグリッドを衝突判定により自動抽出する時間と、すでに可視オブジェクトグリッドを格納したテーブルから、ある視点からの可視オブジェクトグリッドを検索する時間を測定した。

実験方法は、まず 2 次メッシュ (国土地理院) を x, y 方向に各 2000 分割し 3 次メッシュを生成し、3 次元グリッドからオブジェクトグリッド G を抽出し、視点となる 1 つの道路オブジェクトグリッド R のから任意の範囲内 (ここでは視点からグリッド数 20 以内) にある建物オブジェクトグリッド B との x - y 平面での衝突判定を行い、可視オブジェクトグリッド Vis を抽出した (図 5)。この際の可視判定の衝突計算時間は約 0.15sec であった (Intel Pentium4, 1.70GHz, 512MB)。視点となるオブジェクトグリッド数は全体のグリッドに対し約 1 割であったことから、2 次メッシュを x, y 方向で各々 2000 分割したグリッド 2000×2000 の 1 割で 400000 個となり、単純計算では x - y 平面での 2 次メッシュごとの可視・不可視判定時間は $0.15\text{sec} \times 400000 = 60000\text{sec} = 16.7\text{h}$ となる。実際には、相対的位置情報としての建物オブジェクト、場所、地域の考慮や、道路、建物オブジェクト毎に分割数を変えることで計算対象となる有効なオブジェクトグリッド数が削減される。また本稿で提案した可視・不可視判定は視点となるオブジェクトグリッドごとに行うことから、視点ごとに並列処理が行えるため台数効果による大幅な計算時間の短縮が図れる。

一方、可視オブジェクトグリッドテーブルから、図

5と同様の視点からの可視オブジェクトグリッド Vis の検索時間は 0.01sec であった．複合索引のためテーブルのタプル数が膨大になっても瞬時に検索可能なため，本稿の提案手法は事前処理に多少時間がかかっても 1sec あたりに対応できるトランザクション量が多くなり商用のサービスにおいても有効である．

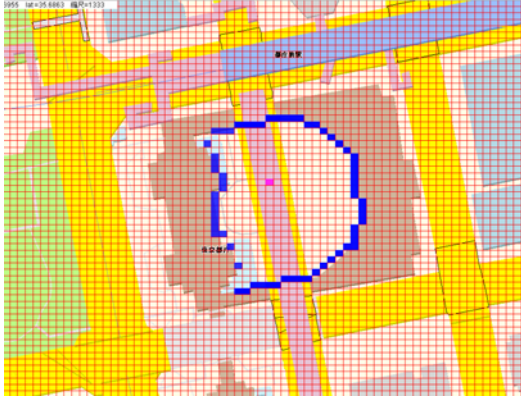


図 5：視点 1 点からの可視判定結果

Fig5.Visible judgment result from aspect one point

4. 相対的位置情報を用いたユーザ位置検索法

4.1. 基本方針

事前計算による索引であるオブジェクトグリッド，可視・不可視判定結果を位置推定計算ロジックにて利用する．その結果，ユーザ位置の候補となるオブジェクトグリッド（以下，候補グリッドとする）のみの計算対象による計算量削減と，建物オブジェクトの可視部分のみのベクトル計算による位置推定精度向上となる．ユーザ位置推定は，ユーザ位置候補となるオブジェクトグリッドの重心にユーザからの相対的位置情報を方向ベクトルとして与え計算する．次節で位置検索アルゴリズムを詳述する．

4.2. 位置検索アルゴリズム

オブジェクトグリッドテーブル，可視オブジェクトグリッドテーブルを用い，建物オブジェクト A，B の相対的位置情報からユーザ位置を推定するグリッドを求めるベクトル計算のアルゴリズムを示す（図 5）．

ここで， $B_A, B_B, VisG_{B_A}, VisG_{B_B}$ を次のように定義する．

- ユーザが視認した建物オブジェクト A，B の建物オブジェクトグリッド集合

$$B_A = \{b_A^i \mid i = 1, \dots, m_A\}$$

$$B_A = \{b_A^i \mid i = 1, \dots, m_A\}$$

- 可視オブジェクトグリッドテーブルによる，オブジェクトグリッド集合 $\{B_A, B_B\}$ を各々可視可能な可視オブジェクトグリッド集合

$$VisG_{B_A} = \{visg_{B_A}^i \mid i = 1, \dots, k_{B_A}\}$$

$$VisG_{B_B} = \{visg_{B_B}^i \mid i = 1, \dots, k_{B_B}\}$$

【アルゴリズム】

1. 建物オブジェクト A, B を共に視認可能な候補グリッド集合 O を求める．

$$O = VisG_{B_A} \cap VisG_{B_B} = \{o_j\}$$

$$(j = 1, \dots, k)$$

以下の 2, 3 を視認した建物オブジェクト A, B に対して繰り返す．

2. O の各候補グリッドからの B_A 中の可視オブジェクトグリッド Vis （グリッド数： l ）を可視オブジェクトグリッドテーブルより抽出し， O の各候補グリッドから可視オブジェクトグリッド

(l) に対するベクトル \vec{V}_{Ai} を計算する．大きさは常に 1 でベクトルの向きは建物オブジェクト A の各オブジェクトグリッドが見えるようなユーザの向きとする．

$$\vec{V}_{Ai} = (\vec{v}_{Ai}) \quad (j = 1, \dots, k)$$

3. O の各候補グリッドが持つ複数のベクトルを加算し，可視オブジェクトグリッド数 (l) で割る．これは，各候補グリッドからの建物の可視オブジェクトグリッド数が異なるため，建物の可視オブジェクトグリッド数が多い建物に合計されたベクトルが左右されないようにするためである．

$$Sum \vec{V}_{Ai} = \frac{(\vec{v}_{A1} + \vec{v}_{A1} + \dots + \vec{v}_{Al})}{l}$$

$$(j = 1, \dots, k)$$

4. O の各候補グリッドについて，候補グリッドが持つ複数のベクトルを加算し，そのベクトルが最大である候補グリッド，またはその集合 W を出力する．

$$W = \{o_j \mid o_j \in O \wedge$$

$$(\forall_h) (|\sum \vec{V}_{Aj} + \sum \vec{V}_{Bj}|) \geq (|\sum \vec{V}_{Ah} + \sum \vec{V}_{Bh}|)$$

【ベクトル計算式】

建物オブジェクト A を構成するオブジェクトグリッド 1 つの重心座標，候補グリッド 1 つの重心座標を次のように定義する．

$$B_A = (x_A, y_A, z_A)$$

$$O = (x_O, y_O, z_O)$$

求めるベクトル \vec{V}_A は次のように示される．

$$\vec{V}_A = (x, y)$$

$$x = \cos \theta * \frac{x_A - x_O}{\sqrt{(x_A - x_O)^2 + (y_A - y_O)^2}} - \sin \theta * \frac{y_A - y_O}{\sqrt{(x_A - x_O)^2 + (y_A - y_O)^2}}$$

$$y = \sin \theta * \frac{x_A - x_O}{\sqrt{(x_A - x_O)^2 + (y_A - y_O)^2}} + \cos \theta * \frac{y_A - y_O}{\sqrt{(x_A - x_O)^2 + (y_A - y_O)^2}}$$

$\theta = 0, 45, 90, 135, 180, 225, 270, 315$ (8 方向の場合)

例えば, 建物が前に見える時は $\theta = 0^\circ$ とする .

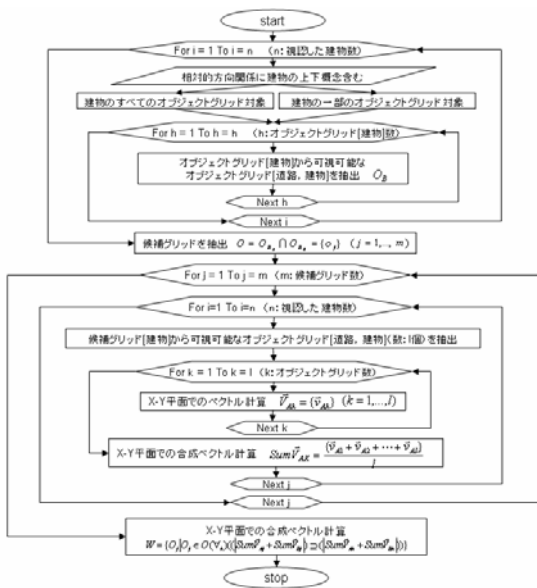


図 5 : 位置検索アルゴリズム

Fig5. Location Identification Algorithm

5. 実装

可視オブジェクトグリッド抽出アルゴリズム，位置検索アルゴリズムを用い実装を行った．

5.1. 実装概略と結果

地図データは道路・建物オブジェクトグリッドを抽出するのに必要なポリゴンデータの道路，建物データが取得可能な，昭文社製の MAPPLE デジタル地図デー

タ MAPPLE10000(MDX フォーマット : バイナリ形式) を使用した .

まず 2 次メッシュを x , y 方向に各 2000 分割し，さらに z 方向に 5m 間隔で分割し，連続空間を 3 次元グリッドを生成し，地図データよりオブジェクトグリッド G を抽出した . 道路，建物オブジェクトは共に ID (X,Y,Z) の Z=0 上に立っていることを前提とし，高さは固定で図 6 のように付与し他の建物オブジェクトを 15m とした . 図 6 は東京都庁周辺の抽出したオブジェクトグリッドの x-y 平面での表示ウィンドウであり，図 7 は 3 次元表示である . 抽出結果をオブジェクトグリッドテーブルに格納した .



図 6 : オブジェクトグリッド抽出結果表示
Fig6. Object Grid Detection Result

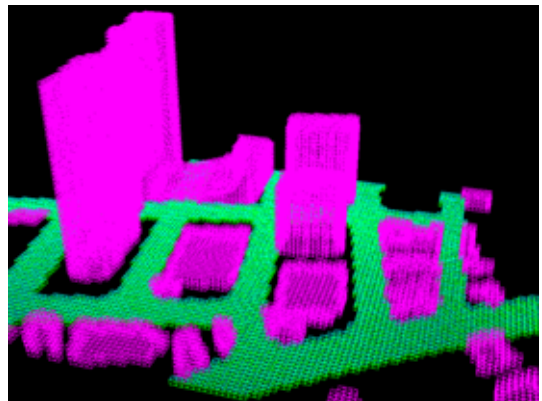


図 7 : オブジェクトグリッド抽出結果 3D 表示
Fig7. Object Grid Detection Result(3D)

次に，視点となる道路オブジェクトグリッド R から任意の範囲内(今回は分割数 20 の MBR 内)にある建物オブジェクトグリッド B の可視・不可視判定を行い，可視オブジェクトグリッド VisG を抽出し，可視オブジェクトグリッドテーブルに格納した . タプル数が膨大となるため複合索引をはった . 図 8 は視点である ID(1148,455,0)からグリッド数 20 の範囲内にあ

るオブジェクトグリッド可視・不可視判定をした結果の x-y 平面の表示ウィンドウであり，図 9 は 3 次元表示である．以上により地図データを索引付けした．



図 8：視点 1 点からの可視判定結果表示
Fig8. Visible Grid Detection From One Standing Position

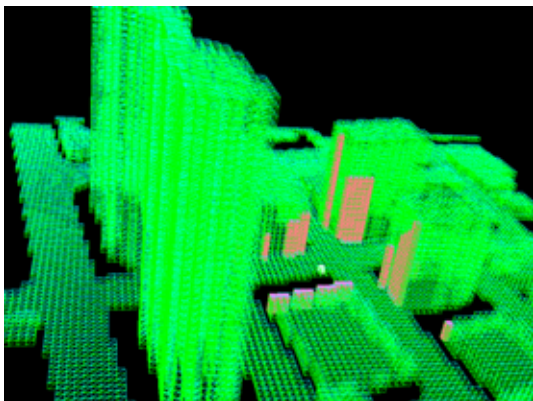


図 9：視点 1 点からの可視判定結果 3D 表示
Fig9. Visible Grid Detection From One Standing Position(3D)

次に，ベクトル計算の対象となる，候補グリッドを取得する．ユーザから相対的位置情報（建物名，相対的方向関係）が入力された際，まず，建物の GeometryID を取得し，オブジェクトグリッドテーブルより建物を構成する建物オブジェクトグリッド B を取得する．次に可視オブジェクトグリッドテーブルより，取得した建物オブジェクトグリッド B が可視可能な道路オブジェクトグリッド R を取得し，視認した建物すべてを可視可能である道路オブジェクトグリッド R を候補グリッドとする．図 10 は建物オブジェクトを時計回りに前，後，左に 3 つ視認した際の候補グリッド抽出結果である．



図 10：候補グリッド抽出結果
Fig10. Grid Candidate Detection Result

その後，視認した建物毎に候補グリッドから可視可能な建物オブジェクトグリッド B を可視オブジェクトグリッドテーブルより取得し，候補グリッドからの視認した建物の可視オブジェクトグリッドへのベクトル計算を行い，図 11 のようにユーザの位置推定を確認した．



図 11：ユーザ位置推定結果
Fig. User Position Presumption Result

6. まとめと今後の課題

相対的位置情報を用いたユーザ位置検索法として，まず現実世界を 3 次元グリッド化し，オブジェクトグリッド G を抽出し，可視・不可視判定を行い $D B$ へ格納することで地図データを事前計算により索引付けした．以上より，ある視点からの可視可能なオブジェクトグリッドが瞬時に取得可能なため，ユーザからの相対的位置情報を用いて，視認した建物オブジェクトがすべて可視可能な候補グリッドを抽出し，各候補グリッドから可視可能なオブジェクトグリッドを用いた，ユーザ位置推定を行なう位置検索アルゴリズムを提案し構築を行った．

今後は，z 方向の曖昧性のモデル化を行い，ユーザ

から視認した建物に対する上下方向概念を導入することで建物内での位置推定を可能とし，さらにより現実モデルにするための標高データの導入，本研究手法の有効性検証のためのモバイル端末などでの実証を目指す．

【謝辞】

本研究を進めるにあたりご助言頂きましたジクー・データシステムズ株式会社開発部長本丸達也氏に謝意を表します．

文 献

- [1] Egenhofer. M: "Spatial Relation: Models, Interfaces, and their Future Application", Proceeding of Advanced Database Symposium '96, Vol.8, No.4, pp.403-424, 1996.
- [2] M, Vazirgiannis "Uncertainty handing in spatial relationships" Proceedings of the 2000 ACM symposium on Applied computing, 2000.
- [3] 上田道 "ウェアラブル環境におけるユーザの相対的奉公関係を用いた位置検索システムの研究" お茶の水女子大学修士論文, 2004.
- [4] 上田道, 佐藤玲奈: "ウェアラブル環境におけるユーザの相対的方向関係を用いた位置検索システムの研究" 情報処理学会研究報告 (DEWS2003), 2003-DBS-131(46), 2003.
- [5] 安田桜, 上田道 "ウェアラブルコンピュータを用いた3次元位置検索システムの構築" お茶の水女子大学 理学部 情報科学科 第9回 卒業研究発表会 要旨集, pp.62-63, 2002.