DEWS2007 E1-5

幾何学的位置関係を用いた類似場所検索の提案

志水　亨†　池崎　正和†　渡邉　豊英†　牛尼　剛聡††

† 名古屋大学大学院情報科学研究科社会システム情報学専攻　〒 464-8603 名古屋市千種区不老町
†† 九州大学芸術工学研究院　〒 815-8540 福岡市南区塩原 4-9-1
E-mail: †{shimizu,mikezaki,watanabe}@watanabe.ss.is.nagoya-u.ac.jp, ††ushiama@design.kyushu-u.ac.jp

あらまし　これまで，地理情報システムや空間データベースの分野において，個々のオブジェクト間の関係を表現する研究が多数行われてきた．これらの既存研究は，2 項間の関係を定義するものが多く，対象となる 2 つのオブジェクト以外に注意を払う研究はなされてこなかった．我々は，与えられたオブジェクト全体に注意を払い，オブジェクト集合が持つ構造のモデル化を目指す．本モデルにより，位相的変化に強い特徴を地図データより抽出でき，2 つのオブジェクト集合に対して類似性の判断が容易になる．本稿では,「重複」・「隣接」・「近傍」の 3 つの幾何学的位置関係を定義し，定義した位置関係を用いてオブジェクト集合が持つ構造をグラフ構造で表現する．さらに，提案モデルの利用例として，類似場所を検索するシステムを提案し，その実装について述べる．
キーワード　場所検索，空間関係，グラフ構造

Similar Location Retrieval Based on Geometrical / Spatial Relations

Toru SHIMIZU†, Masakazu IKEZAKI†, Toyohide WATANABE†, and Taketoshi USHIAMA††

† Department of Systems and Social Informatics, Graduate School of Information Science, Nagoya University
Furo-cho, Chikusa-ku, Nagoya, 464-8603, Japan
†† Faculty of Design, Kyushu University　9-1 Shiobaru 4-chome, Minami-ku, Fukuoka, 815-0032,Japan
E-mail: †{shimizu,mikezaki,watanabe}@watanabe.ss.is.nagoya-u.ac.jp, ††ushiama@design.kyushu-u.ac.jp

Abstract　Up to now, many researchers have proposed spatial relations between geographical objects in fields of geographical information systems and spatial databases. However, these proposals have focused on binary relations to represent a feature of geographical objects, but have not concentrated on a feature of the complexity among geographical objects. In this paper, we aim to model the complex structure among geographical objects as a graph structure. The graph structure is robust over topological transformations, and it is useful to compare two sets of objects. To achieve our objective, we define geometrical / spatial relations such as an overlapping, an adjoining, and a neighboring, and generate a graph structure for connecting geographical objects using these three relations. Additionally, we discuss a retrieval system which finds similar locations by a query map.
Key words　Location retrieval, Spatial relation, Graph structure

1. はじめに

空間データベースや地理情報システムの分野において，オブジェクト間の位置関係の定義・利用が進められている．また，オブジェクト間の位置的・空間的関係の応用として，空間クエリの規定 [2] や，空間データマイニングの提案 [8] がなされてきた．オブジェクト間の位置関係は，数値的な位置関係と位相的な位置関係の 2 つに分類することができる．数値的な位置関係では，距離や方角を用いてオブジェクト間の関係を定義する [9], [11]．距離や方角を利用することで,「駅から 500m の距離にあるコンビニ」や,「駅の北の喫茶店」等の関係が表現可能になる．数値的位置関係の利用例としては，オブジェクト間の距離に注目することで，付近にあるオブジェクトの組合せを発見し，データマイニングに活用する研究が森本によって提案されている [9]．また，方角に注目することで，複雑な形状であってもオブジェクト間の空間関係を表現できることが，Skiadopoulos らによって提案されている [11]．

他方，位相的な位置関係とは，オブジェクトの領域が持つ位相的な性質に注目して定義・表現される関係である．位相的な性質に注目することで,「隣接」や「包含」等の地図上に存在する幾何学的な知識が表現可能になる．代表的な位相的位置関係として，Egenhofer らによって提案された 9-Intersection があ

る [4]．9-Intersection では，オブジェクトの領域間における共通部分に注目し，2 つの領域の外部・境界・内部がそれぞれ共通部分を持つかどうかを判別している．ここで，Florence らが述べているように，9-Intersection を用いて位置関係を定義した場合，求められる関係の多くは Disjoint 関係となる [7]．この問題を解決するために，Chen らによって，ボロノイ図の利用による Disjoint 関係の表現の高度化が提案されている [1]．他にも，より複雑な形状に対して 9-Intersection を適用するような提案がなされている [10]．このように，関係の細分化を進める研究が多い一方で，関係の細分化によって類型化が困難になる問題が挙げられている [3]．

本稿では，関係を細分化して特徴的な関係を定義するのではなく，単純な関係の組合せによるオブジェクト集合が持つ構造の表現を目指す．そして，表現した構造を利用することで，地図データに対する類似場所検索法を提案する．地図データに対する場所検索法として，Egenhofer らによって提案された Query-by-Sketch が著名である [5], [6]．Query-by-Sketch は，ユーザが入力したスケッチから 9-Intersection に基づく位置関係を抽出し，クエリを生成することで，ユーザが描いた場所を発見する手法である．Query-by-Sketch では，オブジェクトが示す領域が離れている場合に関係の定義が単一になる問題を解決するために方向関係を併用している．しかし，方向関係は位相的不変量ではないため，例えば地図データに対して伸縮を加えた場合などに，容易に関係が変化する．加えて，方向関係は曖昧な地図からでは正しく導出できない．本稿では，これらの問題を解決し，より曖昧なスケッチをクエリとした類似場所検索について述べる．

以降，2 章において対象を規定し，類似場所検索の枠組みについて述べる．3 章では位置関係の抽出法及び類似場所検索手法について述べる．そして，4 章において試作した検索システムの概要を説明し，5 章では検索の評価実験について説明し，考察する．最後に，6 章で本稿をまとめる．

2. フレームワーク

2.1 研究概要

本研究の目的は，空間データベース中のオブジェクト集合が持つ構造の表現による，類似場所検索の実現である．図 1 に，本研究の枠組みを示す．本研究では，精度の高い地図データだけでなく，ユーザが作成する曖昧なスケッチも地図データとして扱う．地図データは複数のオブジェクトで構成される．与えられた各オブジェクトに対して位置関係を求め，各オブジェクトをノードに，位置関係をリンクとしてグラフ構造を作成する．オブジェクト間の関係や，オブジェクト集合から求めた関係を用いることで，オブジェクト集合が持つ構造を表現可能になる．得られた特徴を比較することで，2 つのオブジェクト集合の類似性を判断する．本稿では，オブジェクト集合の類似性判断のために，オブジェクト集合が持つ特徴をグラフ構造で表現する．グラフ構造の類似性を判断することで，片方のオブジェクト集合中に存在する他方のオブジェクト集合と類似したオブジェクト集合を発見する．本稿では，類似したオブジェクト集合を検出することで，類似場所検索を実現する．

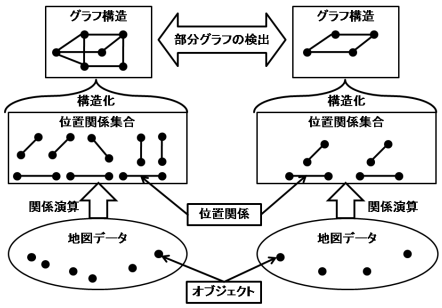

図 1　概念的枠組み
Fig. 1　Conceptual framework.

2.2 形式化

まず，本稿で扱う地図データを定義する．地図データは複数のオブジェクトで構成されるオブジェクト集合とする．一般的な地図データにおいて，地図データ中のオブジェクトは様々な属性値を持つが，分類，オブジェクトの形状，オブジェクトの座標情報のみを用いる．本稿では，オブジェクトは行政区域，建築物，及び河川などとして地図中で表現されるものとする．オブジェクトは以下を満たす 4 項（ID，分類，オブジェクトの形状，オブジェクトの座標情報）で表される．

- ID はオブジェクトを唯一に識別する記号である．
- オブジェクトの分類は地図データに従う．
- オブジェクトの形状は，点，線，多角形のいずれかである．
- オブジェクトの座標情報は，二次元平面上の座標の列である．

本稿では，単一の線分だけでなく，複数の線分から構成される折れ線も線として扱う．したがって，線は複数個の頂点を結ぶことで作成される．多角形は，折れ線によって囲まれた領域を指し，線分と同様に複数個の頂点を結ぶことで作成される．オブジェクトの形状及び座標情報について，以下のように定める．

- 点は単一の座標で表現され，二次元平面上のある一点を示す．
- 線は 2 個以上の座標で表現され，端点を境界として頂点を繋いだ折れ線であることを示す．
- 多角形は 3 個以上の座標で表現され，線分を境界とする閉じられた領域であることを示す．

次に，オブジェクト間の関係として，位置関係を定義する．位置関係は 2 つのオブジェクトの間に存在する 2 項関係であり，以下を満たす 3 項（種類，第 1 オブジェクト，第 2 オブジェクト）として定める．なお，関係の種類は 2.3 節で定める幾何学的位置関係のいずれかとする．

- ある第 1 オブジェクト，第 2 オブジェクトに対して，位置関係は唯一である．
- 第 1 オブジェクト，第 2 オブジェクトは異なるオブジェクトである．

- 第 1 オブジェクト，第 2 オブジェクトは可換である．

最後に，位置関係グラフについて定義する．位置関係グラフは，オブジェクトをノードとし，位置関係をノード間のリンクとして構築される．そして，リンクである位置関係の定義に従い，生成される位置関係グラフはループ及び多重辺を持たない単純グラフとなる．

2.3 幾何学的位置関係

本稿では，関係の組合せを用いてオブジェクト集合の構造を表現するため，個々の関係の多様性よりも単純さを重視し，重複と隣接を規定する．また，位相的位置関係に加えて，オブジェクトのつながりに注目する．オブジェクトのつながりとは，あるオブジェクトの傍に別のオブジェクトが存在している状態を示す．このような性質を近傍とする．本研究では，オブジェクトのつながりを表現するために，ボロノイ図を利用する．以上を踏まえ，位置関係の意味を以下の 3 つに類型化する．

- 重複
- 隣接
- 近傍

重複は，2 つのオブジェクトがその内部において共通部分を持つことを示す．隣接は，2 つのオブジェクトがその境界部分において共通部分を持つことを示す．重複と隣接を表現するために，Overlapping 関係と Adjoining 関係の 2 つを定義する．ここで，重複と隣接では重複を優先する．すなわち，2 つの領域が内部及び境界部分において共通部分を持つ場合には，Overlapping 関係に分類される．図 2 に，Overlapping 関係と Adjoining 関係による類型化を示す．前節で述べたように，オブジェクトの形状としては点，線，多角形を考える．点は単一の座標であり，内部及び境界部分を持たないと考える．したがって，点と点の位相的位置関係を定義しない．線については，その端点を境界部分とし，端点間に形成される複数の線分を内部とする．多角形については，その境界線分を境界部分とし，領域を内部とする．

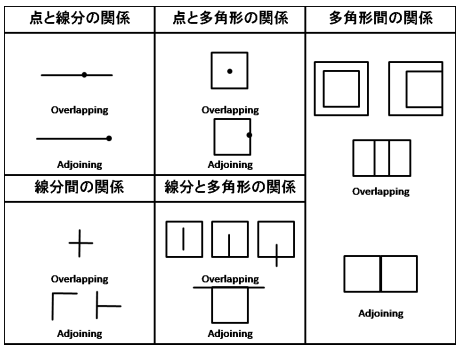

図 2　領域間の位相的位置関係
Fig. 2　Topological relations between regions.

近傍を表現するために Neighboring 関係を定義する．本稿では，ボロノイ図を用いることでオブジェクトのつながりを表現する．対象として与えられたオブジェクト集合に対してボロノイ図を作成することで，ボロノイ領域が隣接するオブジェクトの組を求める．近傍は，2 つのオブジェクト間においてボロノイ領域が隣接することを示す．ただし，2 つのオブジェクトが重複及び隣接を持つ場合には，近傍にはならないとする．

3. 位置関係に基づく類似場所検索

3.1 位置関係演算

本節では，位置関係の演算について述べる．位置関係を求めるために使用する論理関数として，IsOverlapping，IsAdjoining，IsNeighboring を定める．各関数は，引数にオブジェクトの ID を 2 つ持ち，2 つのオブジェクトがそれぞれ対応する関係にあるかどうかを真か偽で返す．Overlapping 関係及び Adjoining 関係は 2 項関係であるため，引数の 2 つのオブジェクトから求めることができる．一方，Neighboring 関係を求めるためには，与えられた地図データよりボロノイ図を作成する必要がある．IsNeighboring 関数は，作成されたボロノイ図に基づいて，引数の 2 つのオブジェクトのボロノイ領域が隣接するかどうかを真か偽かで返す．以下，各論理関数について述べる．

3.1.1 IsOverlapping 関数

IsOverlapping 関数では，領域の内部が共通部分を持つかどうかを検出する．線を対象とする場合，2 線が共通部分を持つためには，2 線が端点以外で交差・接触する必要がある．また，線と多角形が共通部分を持つためには，対象とする線の構成線分が多角形の境界線分と交差する必要がある．交差しない場合，線を構成する頂点もしくは線の構成線分が多角形の内部に含まれる必要がある．多角形を対象とする場合，領域の内部が共通部分を持つためには，片方の頂点もしくは境界線分が他方の内部に含まれる必要がある．また，頂点及び境界線分が内部に含まれない場合でも，2 つの多角形の境界線分が交差する場合，領域の内部が共通部分を持つと言える．

3.1.2 IsAdjoining 関数

IsAdjoining 関数では，領域の境界が共通部分を持つかどうかを検出する．与えられた領域がその内部において共通部分を持たないとき，IsAdjoining 関数を用いて境界が共通部分を持つかどうかを調べる．線を対象とする場合，2 線の境界が共通部分を持つためには，片方の端点が他方の構成線分上に含まれている必要がある．また，線が多角形と接触している場合は，線の構成線分が多角形の境界線分と共通部分を持つ必要がある．多角形を対象とする場合には，片方の境界線分と他方の境界線分が共通部分を持つ場合のみ，2 つの多角形が接触していると言える．

3.1.3 IsNeighboring 関数

IsNeighboring 関数の前処理として，与えられた地図データからボロノイ図を作成する．IsNeighboring 関数は，作成したボロノイ図に基づいて，2 つの領域のボロノイ領域が隣接しているかどうかを判定する．本稿では，離散ボロノイ図の考え方を利用して，一般図形に対するボロノイ図を作成する．離散ボロノイ図は，平面を画素の集合として捉え，各画素を最も近い母点に所属させることで作成する．本稿では，画素に相当するものとして，地図データが示す領域中に微小間隔に点を配置する．さらに，対象とするオブジェクトは領域を持つため，どの母点に属するかではなく，どの領域に属するかを判断する．本

稿では，所属する領域を判定するために，領域を構成する線分との距離を計算する．そして，微小間隔に配置された点は，各点にとって最も近い線分を持つ領域に属するとする．

図 3 に，地図データから描画される地図と，地図データから作成される離散ボロノイ図を示す．図 3 の左側の地図を構成する各領域に対して離散ボロノイ図を構築したものが，右の図となる．右の図では，所属する領域が変化した点を描画した結果，ボロノイ境界が形成されている．

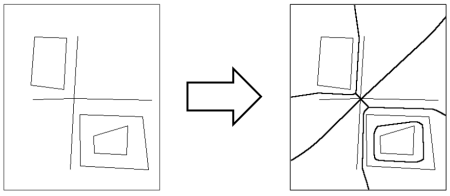

図 3　離散ボロノイ図の例
Fig. 3　Example of discrete Voronoi diagram.

このようにして得られたボロノイ図より，ボロノイ領域が隣接するオブジェクトの組を取得する．IsNeighboring 関数は，与えられた 2 つのオブジェクトが以下の条件を満たすときに真を返す．

- Overlapping 関係及び Adjoining 関係にない
- ボロノイ領域が隣接している

3.2　部分グラフの検出

本節では，類似場所発見のための部分グラフ検出について述べる．提案手法では，検出元として用いるグラフ（以下，検出元グラフ）と，クエリとして用いるグラフ（以下，クエリグラフ）の 2 つを入力とする．そして，クエリグラフの各ノードに対応するノードを検出元グラフから発見する．以下，各ノード集合について定める．まず，検出元グラフのノード集合 C，及びクエリグラフのノード集合 V は以下のように表現される．

$$C = \{c_1, c_2, \ldots, c_n\}$$
$$V = \{v_1, v_2, \ldots, v_m\}$$

これらのノード集合に対して，ノード集合 V の i 番目のノードを取得する関数 $Get(V, i)$ を定める．

次に，グラフ中のリンク l を定める．l は 3 種類の関係のいずれかに分類され，2 つのノードで構成される．先に述べた位置関係をリンクとすることで，ノード n_i に対して，n_i と関係を持つノード集合 $Linked_{n_i}$ が得られる．例えば，検出元グラフ中のノード c_i に対する $Linked_{c_i}$ は以下のように表現される．

$$Linked_{c_i} = \{c | link(c_i, c)\}$$

ここで，$link(c_i, c)$ とは，c_i と c の間に関係が存在することを示す．

次に，検出元グラフのノードのうち，クエリグラフの各ノードと対応するノードの組 $result$ を定める．$result$ は，以下のように表現される．

$$result = (vList, cList | vList[i] \in V, cList[i] \in C.$$
$$cor(vList(i), cList(i)))$$

ここで，$cor(v, c)$ とは，2 つのノード v と c が対応することを示す．$result$ に含まれるノードの組のうち，v に対応する c を取得する関数として，$GetValue(result, v)$ を定める．提案手法では，$result$ の集合 $RESULT$ を最終的な出力として返す．$RESULT$ は以下のように表現される．

$$RESULT = \{result_1, result_2, \ldots, result_n\}$$

以上の表現を用いて，部分グラフ検出に用いる再帰関数 $UNIFY$ を説明する．図 4 に $UNIFY$ のアルゴリズムを示し，部分グラフ検出の手順について述べる．

```
UNIFY(v_i, C', result)
 1  if ♯result = ♯V
 2     then result を RESULT に追加
 3     else
 4          for j ← 0 to ♯C'
 5          c_j ← Get(C', j)
 6          if IsMatching(v_i, c_j) = TRUE
 7            then result に (v_i, c_j) を追加
 8                 NextVList ← RemoveCheckedNode(Linked_{v_i}, result)
 9                 if ♯NextVList ≠ 0
10                   then v_next ← NextVList_0
11                        C'Next ← GetLinkedNode(c_j)
12                        goto line 24
13                   else
14                        for p ← 0 to ♯V
15                        v_p ← Get(V, p)
16                        if v_p ∉ vList
17                          then v_next ← v_q
18                               for q ← 0 to ♯V
19                               v_q ← Get(V, q)
20                               if v_q ∈ vList and v_q ∈ Linked_{v_next}
21                                 then c_r ← GetValue(v_q, result)
22                                      C'Next ← GetLinkedNode(c_r)
23                                      goto line 24
24                 UNIFY(v_next, C'Next, result)
```

図 4　部分グラフ検出アルゴリズム
Fig. 4　Algorithm for subgraph detection.

起点とするノード v_{start} をクエリグラフの任意のノードから選択する．v_{start} に対応するノード集合を C より選択し，C' とする．対応したノードの組を格納するために，空の $result$ を作成する．v_{start}，C'，及び $result$ を入力として，関数 $UNIFY$ を実行する．$UNIFY$ によって得られた $result$ の集合 $RESULT$ が，検出結果となる．

$UNIFY$ は，マッチングするノード v_i，v_i に対応する可能性のあるノード集合 C'，及び対応するノードの組 $result$ を入力とする．C' に含まれる各ノードが v_i に対応するか調べ，対応するならば $result$ に追加する．対応を調べるために，$IsMatching$ 関数を用いる．$IsMatching$ 関数では，クエリグラフ中のリンク及び検出元グラフ中のリンクを利用して，引数の 2 つのノードが対応するか調べる．

対応した場合，再帰の際に利用するノード v_{next} と，

v_{next} に対応する可能性のあるノード集合 $C'Next$ を求める．$UNIFY$ では，v_i のリンクノード $Linked_{v_i}$ から一つ選び，次に対応させるノード v_{next} とする．ここで，$Linked_{v_i}$ に対して関数 $RemoveCheckedNode$ を適用することで，$Linked_{v_i}$ から既に対応済みのノードを削減する．$RemoveCheckedNode(Linked_{v_i}, result)$ の結果を $NextVList$ に代入し，$NextVList$ の要素数を条件にして分岐する．

$NextVList$ の要素数が 0 でない場合，すなわち，v_i に未対応のリンクノードが存在する場合は，$NextVList$ から一つノードを選び，v_{next} とする．v_{next} に対応する可能性のあるノード集合 $C'Next$ は，v_i に対応したノード c_j を入力とする関数 $GetLinkedNode$ を用いて求める．$GetLinkedNode(c_j)$ は，$Linked_{c_j}$ を返す．そして，v_{next}，$C'Next$，及び $result$ を入力として，再帰する．

$NextVList$ の要素数が 0 の場合，すなわち，v_i のリンクノードは全て対応済みの場合は，V の中で未対応のノードを探し，v_{next} とする．さらに，V の中で対応済みかつ，$Linked_{v_{next}}$ に含まれるもの v_q を探す．v_q に対応しているノード c_r を $GetValue$ 関数を用いて $result$ より取得し，c_r のリンクノード集合を求め，$C'Next$ とする．こうして得られた v_{next}，$C'Next$，及び $result$ を入力として，再帰する．

$UNIFY$ は，クエリグラフの全てのノードを対応した場合に，得られた $result$ を $RESULT$ に追加する．v_i に対して，対応する可能性のある全ての c'_j を対応させることで，すべての処理を終了し，$RESULT$ を出力する．検索結果として必要となるのは，対応したノードの組ではなく，クエリグラフと類似した構造を持つオブジェクトの集合である．そこで，検索結果を作成するために，$RESULT$ の各要素に対し，$cList$ を取得する．ここで，$cList$ が同一のものは，同一の解として扱う．そして，$cList$ の集合を検索結果として返す．

3.3 対応度と類似度

本節では，グラフ構造間の類似性を評価するための対応度と類似度について述べる．先に述べた部分グラフ検出アルゴリズムでは，2 つのグラフにおいて，関係が一対一で対応したグラフ構造のみが検出される．しかし，スケッチなどの曖昧な地図から作成されたグラフ構造をクエリとする場合，クエリグラフの関係と検出元グラフの関係が一対一で対応しない場合がある．また，地図データに対して位相的変形が加えられた場合においても，関係が一対一で対応しない場合がある．そこで本研究では，2 つのグラフに存在する関係が，どのように対応するか調べ，対応度と類似度を定義する．

Overlapping 関係及び Adjoining 関係は，位相的位置関係であるため，地図データに対して位相的変形が加えられた場合において変化しない．また，オブジェクトの省略や欠損があった場合においても，省略されたオブジェクトで構成される関係が消えるだけである．したがって，Overlapping 関係及び Adjoining 関係は，一対一で対応するか，対応する関係が存在しないかのみとなる．

一方，Neighboring 関係においては，関係は一対多で対応する場合がある．オブジェクトの省略，欠損によって，Overlapping 関係及び Adjoining 関係が存在しなくなり，新しい Neighboring 関係が生まれる．他にも，ボロノイ図の変形によって，それまで隣接していたボロノイ領域の間に他のボロノイ領域が侵入する場合がある．他のボロノイ領域が侵入することで，Neighboring 関係が 2 つに分割される．

Neighboring 関係が一対多で対応する場合のために，対応する関係の個数に基づいて関係間の対応度を定める．関係の対応度は，関数 $count(Relation)$ を用いて求める．$count(Relation)$ はクエリグラフ中の関係が，検出元グラフ中の何個の関係と対応するかによって求められる．まず，クエリグラフ中の i 番目の Neighboring 関係 R_i に対応する関係を調べ，その個数を数える．R_i が，2 つのオブジェクト A，B を構成要素に持つ時，$R_i(Neighboring, A, B)$ とする．与えられた関係 $R_i(Neighboring, A, B)$ に対して，$R(T, A, N_1), R(T, N_1, N_2), \ldots, R(T, N_m, B)$ を満たす最小の m を検出元グラフより算出する．なお，T は，Overlapping，Adjoining，Neighboring の三種類のいずれかとなる．1 を $m+1$ で割り，関係間の対応度とする．ここで，$m = 0$ の時は一対一で対応する場合となるため，関係の種類 T が Neighboring の場合のみ対応するとする．

関係間の対応度を用いて，グラフ構造間の類似度を算出する．式 (1) にグラフ構造間の類似度を算出する式を示す．グラフ構造間の類似度は，関係の対応度の平均として求められる．なお，式 (1) において，N はクエリグラフ中の関係の総数とする．

$$SimilarityDegree = \frac{\sum_{k=1}^{N} count(R_k)}{N} \quad (1)$$

図 5 中の検出元グラフとクエリグラフを用いて類似度計算の例を示す．なお，図 5 中の全てのリンクは Neighboring 関係を示すものとする．クエリグラフ中のノード a，ノード b，及びノード c が，それぞれ検出元グラフ中のノード A，ノード B，及びノード C と対応する場合の類似度を計算する．$R(Neighboring, a, b)$ 及び $R(Neighboring, a, c)$ は，検出元グラフ中にも対応する関係 $R(Neighboring, A, B)$ 及び $R(Neighboring, A, C)$ が存在する．したがって，それぞれの関係の対応度は 1 となる．しかし，$R(Neighboring, b, c)$ に対応する関係である $R(Neighboring, B, C)$ は，検出元グラフ中には存在しない．ここで，$R(Neighboring, b, c)$ は，検出元グラフでは，$R(Neighboring, B, D)$ 及び $R(Neighboring, C, D)$ に対応すると判断する．したがって，対応する関係の個数は 2 となる．同様に個数を数えることで，最も小さい値が算出できる．この例では，最小の対応する関係の個数は 2 であり，$R(Neighboring, b, c)$ の対応度は 0.5 となる．以上より，$SimilarityDegree$ は，$(1 + 1 + 0.5)/3 = 0.83$ となる．

3.4 部分グラフ検出における対応度と類似度の利用

先に述べた部分グラフ検出アルゴリズムに対して，対応度と類似度を適用する．本稿では，$GetLinkedNode$ 関数に対して対応度を適用する．$GetLinkedNode$ は，引数で与えられるノード n と関係を持つノード集合 $Linked_n$ を返す．関係を持つノードの適用範囲を拡張することで，$GetLinkedNode$ を拡

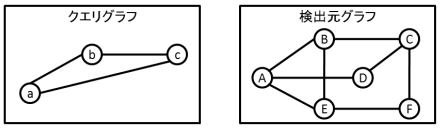

図 5　類似度計算の例
Fig. 5　Example for Similarity degree.

張する．

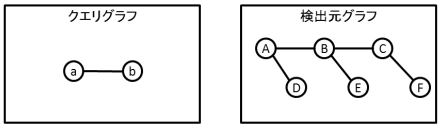

図 6　*GetLinkedNode* 関数と対応度
Fig. 6　Example for *GetLinkedNode* function

図 6 中のクエリグラフと検出元グラフを例として示す．なお，図 6 中の関係は全て同じであるとする．ノード a とノード b で構成される関係 $R(T, a, b)$ の対応度を求める．T が Overlapping か Adjoining であるとすると，$R(T, A, B)$ 及び $R(T, A, D)$ の 2 組が対応度 1 で対応する．T が Neighboring の場合も同様に，$R(T, A, B)$ 及び $R(T, A, D)$ が対応度 1 で対応する．しかし，T が Neighboring の場合には，上記の 2 組以外に，$R(T, A, C)$ 及び $R(T, A, E)$ が対応度 0.5 で，$R(T, A, F)$ が対応度 0.33 でそれぞれ対応する．

これに対し，ノード A と対応するノード集合を返す関数 $GetLinkedNode(A)$ は $\{B, D\}$ を返す．これは，対応度 1 で関係を持つノードの集合である．したがって，例えば対応度 0.5 で関係を持つノード集合は，$\{B, C, D, E\}$ となる．

以上を踏まえ，*GetLinkedNode* 関数を拡張する．新しい *GetLinkedNode* 関数は，引数にノードだけでなく，間に挟んで良いリンクの数を持つ．例えば，$GetLinkedNode(A, 2)$ は，$\{B, C, D, E\}$ を返す．新しい *GetLinkedNode* 関数を用いることで，先に述べたアルゴリズムを拡張する．

先に述べた部分グラフ検出アルゴリズムは，クエリグラフと検出元グラフにおいて，対応するノードの組を比較結果として返す．図 5 の例では，{(A,a)，(B,b)，(C,c)} が出力の 1 つとなる．したがって，対応するノードが異なる場合，例えば {(A,a)，(B,b)，(C,c)} と {(A,b)，(B,c)，(C,a)} とは区別される．これに対し，検索結果はクエリグラフの構造を満たすオブジェクト集合である．したがって，{(A,a)，(B,b)，(C,c)} と {(A,b)，(B,c)，(C,a)} とは同じ解となる．ここで，類似度は対応するノードの組に依存する問題がある．そこで本稿では，解に対する類似度として，同一の解を持つ類似度について最大値を求め，最大類似度として定める．

4. プロトタイプシステム

提案手法に基づいて作成される位置関係グラフを用いた類似場所検索システムのプロトタイプを実装した．開発言語には Java の開発環境である JDK^{TM} 5.0 を利用した．また，地図データとしては，国土地理院が発行している数値地図 2500 を利用した．数値地図 2500 では，地理オブジェクトを 8 つに大別し，合計で 32 個の項目を設けている．このうち，都道府県界や市区町村界等の領域を区分するオブジェクト，及び三角点などの地点を示すオブジェクトを除外し，6 種類 14 項目を利用した．本研究で利用する地理オブジェクトの分類，項目，及びオブジェクトの形状を表 1 に示す．

表 1　地理オブジェクトの種類一覧
Table 1　Type of geographical objects.

分類	項目	オブジェクトの形状
交通施設	鉄道	線
	道路線	線
行政区域	市区町村	多角形
	大字・町丁目	多角形
街区	街区	多角形
公園等場地	鉄道敷	多角形
	都市公園	多角形
	学校敷地	多角形
	神社・寺院境内	多角形
	墓地	多角形
	その他の場地	多角形
内水面	河川	多角形
	湖池等	多角形
建物	公共建物	多角形

図 7 に，プロトタイプシステムの全体像を示す．プロトタイプシステムは，スケッチ作成機構，位置関係導出機構，部分グラフ検出機構，及び地図データ表示機構の 4 つで構成される．また，地図データファイル及び位置関係ファイルを検出元として用いる．位置関係ファイルは位置関係導出機構を用いて地図データファイルより作成する．ユーザは，スケッチ作成機構を用いてクエリとするスケッチを作成する．作成されたスケッチを用いて，位置関係導出機構は位置関係を求め，グラフ構造を作成する．部分グラフ検出機構は，作成されたグラフ構造と位置関係ファイルから作成されるグラフとを比較し，部分グラフの検出を行う．出力された部分グラフを構成する各オブジェクトを地図データファイルより選び，地図データ表示機構は検索結果を地図として出力する．

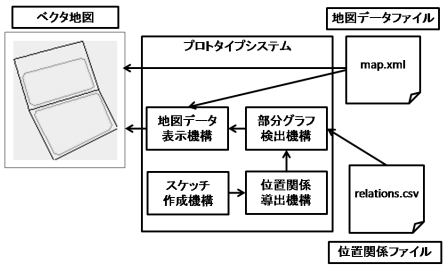

図 7　プロトタイプシステムの全体像
Fig. 7　Overview of the trial system.

図 8 にプロトタイプシステムのインタフェースを示す．図 8

において，画面全体に表示されている地図が，検出元として与えられた地図データに基づいて描画されたものである．また，検索には図中左下に表示されているダイアログを用い，スケッチを作成する．スケッチを作成する際には，まず，ダイアログ左部のボタンを操作することで描画するオブジェクトの種類を選択する．描画するオブジェクトの種類を決定した後，ダイアログ中央をクリックすることで，描画するオブジェクトの領域の頂点を決定するシステムは，入力されたスケッチからグラフ構造を作成し，クエリグラフとする．そして，検出元として与えられる検出元グラフより，クエリグラフと類似した場所を発見する．得られた結果は最大類似度を用いて順序づけされる．

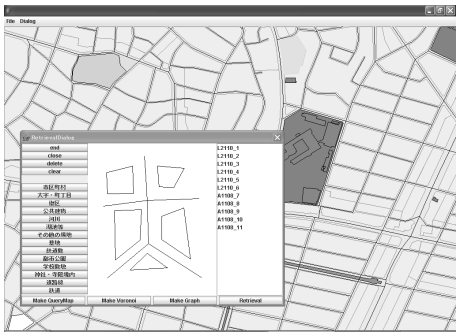

図 8　プロトタイプシステムのインタフェース
Fig. 8　Interface of the trial system.

5. 実　　験

本節では，提案手法を評価するための実験について述べる．プロトタイプシステムのインタフェースを用いて作成したスケッチをクエリとして地図検索を行い，検索結果が妥当なものであるか確認する．実験の手順を以下に示す．

（1） プロトタイプシステムを利用し，対象図葉 M 中の任意の場所 T を元にスケッチ S を作成する．
（2） S からグラフ構造 G を作成する．
（3） M から作成されたグラフと G を比較する．

検索元とする地図データは，数値地図 2500 のうち，07ND151，07ND152，07ND153，及び 07ND154 の 4 図葉を使用した．それぞれの図葉に対し，5 つずつスケッチを作成し，クエリとした．各スケッチをクエリとし，対象とした図葉の地図データを検出元とした．なお，図葉番号は国土地理院が発行している数値地図に割り当てられる番号であり，全国で一意である．

表 2 に検索に用いたスケッチの情報をまとめる．スケッチ番号は，検出元とした図葉に対して，何番目のスケッチであるかを示す．オブジェクト数，位相関係数，及び近傍関係数は，それぞれスケッチ中のオブジェクト数，位相関係数，及び近傍関係数を示す．

表 3 に検索結果をまとめる．スケッチ番号は，表 2 と同様に，何番目のスケッチであるかを示す．処理時間は，検索に要した時間を単位をミリ秒として示す．出現順位は，T が解として確認された順位を示す．類似度は，T が解として確認された際の類似度を示す．

表 2　スケッチの情報
Table 2　Information of sketches.

スケッチ番号	オブジェクト数	位相関係数	近傍関係数
07ND151-1	7	7	6
07ND151-2	9	14	6
07ND151-3	5	3	4
07ND151-4	6	6	5
07ND151-5	3	0	2
07ND152-1	6	7	3
07ND152-2	5	3	4
07ND152-3	3	1	2
07ND152-4	3	0	3
07ND152-5	3	1	2
07ND153-1	7	7	4
07ND153-2	5	4	4
07ND153-3	3	0	2
07ND153-4	5	7	2
07ND153-5	8	10	8
07ND154-1	4	1	2
07ND154-2	5	3	5
07ND154-3	6	6	4
07ND154-4	6	6	4
07ND154-5	6	6	4

表 3　検 索 結 果
Table 3　Experimental result.

スケッチ番号	処理時間 (ms)	出現順位	類似度
07ND151-1	37203	1	1.000
07ND151-2	179110	1	1.000
07ND151-3	664703	45	0.761
07ND151-4	58672	1	0.909
07ND151-5	2672	2	0.750
07ND152-1	27515	1	1.000
07ND152-2	2955781	1	1.000
07ND152-3	3687	1	1.000
07ND152-4	17688	1	1.000
07ND152-5	516	1	1.000
07ND153-1	161172	1	1.000
07ND153-2	38969	7	0.832
07ND153-3	15719	1	1.000
07ND153-4	49953	1	1.000
07ND153-5	228141	1	0.870
07ND154-1	656	1	1.000
07ND154-2	68953	1	0.938
07ND154-3	65281	1	0.900
07ND154-4	422547	5	0.833
07ND154-5	80297	1	1.000

表 3 より，関係数の増加によって処理時間が増加していることがわかる．しかし，07ND152-2 の場合，オブジェクト数，及び関係数が少ないにも関わらず，処理時間は多い．これは，07ND152-2 中の関係の多くは Neighboring 関係であり，一対一で関係が対応しない場合によって組合せが増えているからである．提案手法において，Overlapping 関係及び Adjoining 関

係は，一対一で対応する場合しか考慮しない．したがって，部分グラフ検出時に枝刈りが可能となる．枝刈りの結果，出力される解の個数も減少する．一方，Neighboring 関係は，スケッチの曖昧性を吸収するために，一対多での対応を考慮している．したがって，スケッチ中の関係のうち，Neighboring 関係が増加するに伴って，探索する組合せが増え，処理時間が増大する．また，Neighboring 関係が多い場合，より曖昧な結果も解として出力しやすい．

次に，表 3 の出現順位と類似度について考察する．全体的に，出現順位は 1 位が多く，類似度も高いものが多かった．一方で，いくつかの試行において，出現順位及び類似度は低い値を示した．これは，試行に用いたスケッチにオブジェクトの省略があり，かつ，省略されたスケッチに類似した別の場所が存在したためであった．また，オブジェクトが省略された結果，オブジェクトが統合される場合が確認された．そして，オブジェクトの統合によって類似度が低下することが確認された．図 9 に 07ND153-3 のスケッチとスケッチを作成する際に意図した場所を示す．意図した場所は，図中上部の川の傍に十字路とT字路が存在し，さらに学校敷地が存在する付近であった．これに対し，スケッチには十字路が省略され，川の傍のT字路と学校敷地のみが描かれている．

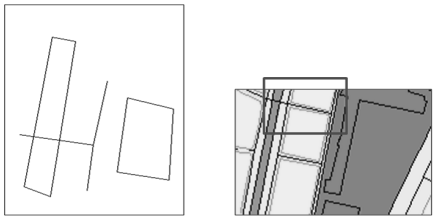

図 9　スケッチと意図した場所の例
Fig. 9　Example of a sketche and .

図 10 に，図 9 のスケッチをクエリとした場合の検索結果を示す．図中左にはユーザが意図しなかった解であるが，スケッチによく似た場所が示されている．一方で，図中右の解はユーザが意図した場所であるが，オブジェクトの省略及び統合によって類似度は低いものとなった．このように，オブジェクトの省略及び統合によってユーザが意図した場所の類似度が低下し，出現順位も低下することが確認された．

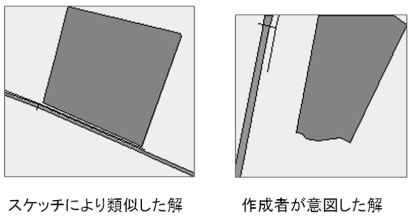

図 10　スケッチにより類似した解と意図した解
Fig. 10　Example of results.

6. ま と め

本稿では，オブジェクト集合が持つ構造のモデル化を目指した．提案モデルでは，オブジェクトをノードとし，オブジェクト間の関係をリンクとしてグラフ構造を作成する．さらに，提案モデルに基づいて類似場所検索システムのプロトタイプを試作した．プロトタイプシステムでは，ユーザが作成したスケッチを入力とし，検出元とする地図データと比較することで，類似場所検索を実現した．また，プロトタイプシステムを用いて評価実験を行うことで，提案モデルが類似場所検索に有効であることを示した．

今後の課題として，オブジェクトの省略によって発生する，オブジェクトの統合への対処が挙げられる．また，部分グラフ検出問題は NP 完全問題であるため，検索にかかる時間はオブジェクト数及び関係数の増加に伴い膨大なものとなる．したがって，検索の高速化のためには効率の良い枝狩りが必要となる．

謝辞　本研究の一部は大幸財団の研究助成，及び名古屋大学 21 世紀 COE プログラム「社会情報基盤のための音声・映像の知的統合」によって実施された．

文　献

[1] Jun Chen, Chengming Li, Zhilin Li, and Christopher M. Gold. A voronoi-based 9-intersection model for spatial relations. *International Journal of Geographical Information Science*, Vol. 15, No. 3, pp. 201–220, 2001.

[2] Eliseo Clementini and Paolino Di Felice. Spatial operators. *SIGMOD Rec.*, Vol. 29, No. 3, pp. 31–38, 2000.

[3] Eliseo Clementini, Paolino Di Felice, and Peter van Oosterom. A small set of formal topological relationships suitable for end-user interaction. In *SSD '93: Proceedings of the Third International Symposium on Advances in Spatial Databases*, pp. 277–295, London, UK, 1993.

[4] M. Egenhofer, D. Mark, and J. Herring. The 9-intersection: Formalism and its use for natural language spatial predicates, May 1994. NCGIA Technical Report 94-1.

[5] Max J. Egenhofer. Spatial-query-by-sketch. In *IEEE Symposium on Visual Languages*, pp. 60–67, Los Alamitos, CA, USA, 1996.

[6] Max J. Egenhofer. Query processing in spatial-query-by-sketch. *Journal of Visual Languages and Computing*, Vol. 8, No. 4, pp. 403–424, 1997.

[7] John Florence and Max J. Egenhofer. Distribution of topological relations in geographic datasets. In *ACSM/ASPRS Conference*, Apr 1996.

[8] Krzysztof Koperski, Junas Adhikary, and Jiawei Han. Spatial data mining: Progress and challenges, 1996. In SIGMOD'96 Workshop on Research Issues on Data Mining and Knowledge Discovery (DMKD'96).

[9] Yasuhiko Morimoto. Mining frequent neighboring class sets in spatial databases. In *KDD '01: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 353–358, New York, NY, USA, 2001. ACM Press.

[10] Markus Schneider and Thomas Behr. Topological relationships between complex spatial objects. *ACM Trans. Database Syst.*, Vol. 31, No. 1, pp. 39–81, 2006.

[11] Spiros Skiadopoulos, Christos Giannoukos, Nikos Sarkas, Panos Vassiliadis, Timos Sellis, and Manolis Koubarakis. Computing and managing cardinal direction relations. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 17, No. 12, pp. 1610–1623, 2005.