位相空間データモデル Universe での空間，時間，時空間データ表現

黒木 進† 牧之内 顕文†

時間的に変化する空間データを時空間データという．従来，時間的に不連続に変化する時空間データモデルが研究されてきたが，現在では時間に関して連続的に変化する時空間データのためのデータベースモデルが求められている．この課題を解決するため，我々は位相空間データモデル Universe を提案する．位相空間データモデル Universe は，時間データと空間データと時空間データを統一して表現・検索することを目的として，これら3種のデータを位相空間という形で抽象化する．そして Universe は抽象化された位相空間を表現・検索するためのデータ表現と演算子を提供する．提案されたデータ表現と演算子を説明すると同時に，時空間データベースへの応用について述べる．

Representation of Spatial, Temporal, and Spatio-temporal Data in the Topological Space Data Model Universe

Susumu Kuroki† and Akifumi Makinouchi†

Spatio-temporal data are such spatial data as change over time. Data models for spatial databases have been studied and presented. In these models, no time-dimension is provided, so objects are supposed to change neither their shapes, nor their positions. Recently, spatial data models with time-dimension, called spatio-temporal data models, have been requested for modelling real world where objects change their positions and shapes over time. This paper presents such a data model based on a mathematical notion to represent gradual changes of objects. This model, named Universe, transforms spatio-temporal data into topological spaces. Universe supports a set of operators and predicates on the topological spaces so that users can retrieve any time-varying data in a unifom way. How to represent spatio-temporal data in Universe and how to use the operators provided by Universe for spatio-temporal queries are discussed in this paper.

1. はじめに

空間データベースにおいて最近関心を集めている研究課題の1つは，空間データが時間的に滑らかに変化していくプロセスをモデル化することである．この課題を解決するために，空間データの時間的変動を表現するデータ（時空間データと呼ぶ）を記録するためのデータモデルの研究が行われている[2),3),8),9),12),14)~17),22)]．また，取り扱うデータの高次元化も課題となっており，一部の画像データベース[11)]のように，画像を特徴付ける統計量によって高次元の図形化されたデータを取り扱う手法もさかんに行われている．また，通常は図形とは考えられていないデータ，たとえば従業員の給料の時間的変化を記述する時間データベースのモデリングに幾何学的な考えを導入し，図形として取り扱うことも考えられる．

このように，空間データの滑らかな時間的変動や，より高次元の図形として表現されるデータや，従来は図形としては取り扱われなかったデータを表現する図形が取り扱えるように従来の空間データ表現を拡張することが必要とされている．この課題を解決するために我々は位相空間データモデル Universe の開発に取り組んでいる[12)]が，ここではこのモデル Universe で図形をどのように表現するか，図形に対する操作としてどのようなものを定義するかを述べる．また，Universe で用いられる図形表現の利害得失について論じる．同時に，このデータモデル Universe を用いた時空間データの表現と検索の例をあげる．

2. 位相空間データモデル Universe

空間データ，時間データ，時空間データを統一的に

† 九州大学大学院システム情報科学研究科
Graduate School of Information Science and Electrical Engineering, Kyushu University

表現・検索するために，我々は位相空間という概念を用いる．位相空間を表現するためのデータモデル Universe で，どのようにこれらデータを表現するか，またどのような演算子を用いて位相空間を検索するか説明する．

2.1 位相空間

位相空間とは三角形や四面体などの図形を抽象化した概念であり，点列 x_n が点 x に限りなく近づくかどうかや，そこで定義された実数値関数が連続かどうかを判定できるような構造を持った集合である．数学的には，位相の定められた集合 X を位相空間という[10]．位相とは開集合系 O によって X 上に決められる構造である．この開集合系 O によって点の近接性や関数の連続性を判定できる．また，位相空間は距離の概念を用いずに定義できるため，さまざまな空間の図形を統一的に取り扱うことができる．

空間データはユークリッド空間の図形を表現するデータである．ユークリッド空間は位相空間であると同時にユークリッド距離を持っているため，位相空間の特殊なケースとして取り扱われる．たとえば，ユークリッド空間に対して，ユークリッド距離を用いて位相空間の面積や体積といった特徴量を計算できる．同時に，そのユークリッド空間は位相空間でもあるので連結成分の数や穴の数などの特徴量も計算できる．

位相空間は図形という概念を抽象化したものであるので，ユークリッド空間に存在する図形だけでなく，仮想的な図形を使って表現される概念についても位相空間を用いて表現することができる．時空間データの表現がその例である．たとえば，三次元ユークリッド空間を並進運動している三次元図形 A を表現するときに，ユークリッド空間と時間との直積からなる四次元空間を定義する．その四次元空間での領域を用いて三次元図形 A の通過した領域を時空間データとして表現する方法がある．運動する物体を位相空間として表現するこの方法は，Space-Time Approach[4),7)]と呼ばれ，コンピュータアニメーションやロボティックスで用いられる．この方法で定義される領域は実在しない仮想的な図形であるが，この仮想的な図形を用いて三次元図形 A の位置と形状の時間的変化を記述できる（**図 1**）．ただし，この四次元空間それ自体に距離を定義することはできず，時間軸においてや，時刻一定を表す等時刻面においては距離を定義できる．

このことは時間データについても成り立つ．時間データとは，ここでは時間的に変化する非空間データである．たとえば，ある地点の温度の変遷は時間データである．先に述べた Space-Time Approach を用いれば，温度の時間的変遷は時間と温度の直積空間において『折れ線』という位相図形に変換される．時間と温度の直積空間においても，先の時空間データの場合と同様，温度差や時間間隔を計算できる．しかし，この位相空間全体に関する合理的な距離は定義できない．

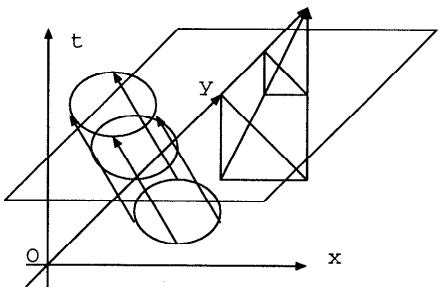

図 1 Space-Time Approach による空間データの時間変化の表現
Fig. 1 Representation of temporal changes of spatial data by the Space-Time Approach.

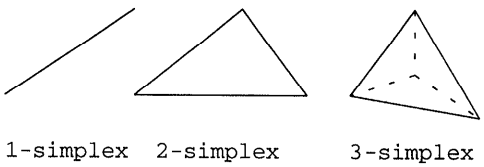

図 2 単体の例
Fig. 2 Simplices.

このように，位相空間という抽象的な概念を用いると，地図データや CAD データはユークリッド空間という特殊な位相空間としてモデル化される．時空間データや時間データを Space-Time Approach で変換することによって得られる仮想的な図形もまったく同様に位相空間として表現できる．それゆえ，従来の空間データ表現の手法を位相空間も取り扱えるように拡張することによって統一的なデータモデルを定義できる．

2.2 位相空間の表現[10)]

位相空間を表現するために，組合せ構造を持った基本的図形として単体（simplex）を定義する．N 次元数空間 R^N の中に一般的な位置にある $n+1$ 個の点 $a_0, a_1, \ldots, a_n$☆をとる．このとき，$\lambda_0+\lambda_1+\ldots+\lambda_n = 1, 0 \leq \lambda_i \ (0 \leq i \leq n)$ のような $n+1$ 個の実数 $\lambda_0, \lambda_1, \ldots, \lambda_n$ によって，$\lambda_0 a_0 + \lambda_1 a_1 + \ldots + \lambda_n a_n$ で表される点全体からなる凸集合を単体と呼び，たとえば σ と表記する．特に次元を明記するときは n 単体と呼ばれる（**図 2**）．この定義から，0-単体は点，1-単体は線分，2-単体は三角形，3-単体は四面体となる．

☆ ベクトル $a_1 - a_0, a_2 - a_0, \ldots, a_n - a_0$ が一次独立．

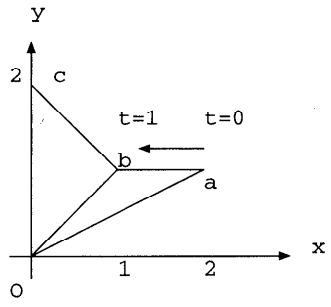

図3 時間変化する2次元空間データ
Fig. 3 Time-varying two dimensional spatial data.

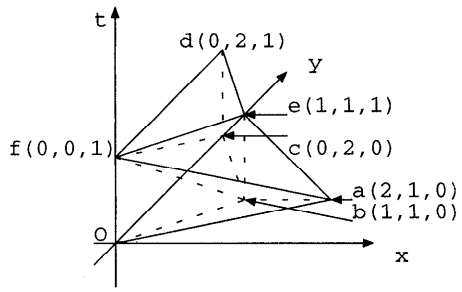

図4 単体的複体の例
Fig. 4 Simplicial complexes.

また，この式で $\lambda_i = 0$ となる点の全体は，a_i を除いた n 個の点を頂点とする単体である．これは単体 σ の辺単体と呼ばれる．辺単体の辺単体もまた辺単体であると定義する．四面体の境界をなす三角形は3-単体の辺単体（face）である．同時に，その三角形の辺単体となっている線分もまた，四面体の辺単体である．

位相空間は必ずしも単体とは限らないので，位相空間を表現するためには基本的図形である単体を組み合わせて，より複雑な図形を定義する必要がある．そこで，基本的図形である単体を元とする集合を考える．このとき単体的複体（simplicial complex）K は次の条件を満たす R^N 中の有限個の単体を元とする集合である．

$\sigma, \tau \in K \Rightarrow$

$\sigma \cap \tau$ は空集合，または σ と τ の両方の辺単体

ここで，時間変化する2次元空間データ（図3）の単体的複体での表現例を示す．

今，2次元空間に2枚の三角面（2単体）Oab と Obc からなる四角面を考える．この四角面は単体的複体として表現されている．いま，三角面 Obc は静止しているが，三角面 Oab が線分 Ob に向かって $t = 0$ から1までの時間にわたって縮小していく過程をモデリングすることを考える．先に述べた Space-Time Approach を用いると，三角面 Obc が時刻0から1にわたってつくる位相空間は三角柱 Obcfed になる（図4）．

線分 Ob へ時刻 $t = 0$ から1にわたって三角面 Oab の各点が近づく速度は，点ごとに異なる．たとえば点 O に近い点での速さは0に近く，点 a の速さが最も大きい．そのため，辺 Oa の動きによって定義される面 Oaef は平面とはならず曲面となる．このことは，4点 Oaef が同一平面上に存在しえないことからも分かる．この曲面を2枚の三角面で近似すると，三角面 Oab がその動きによって定義する位相空間は，四角錐 Obefa となる．この2つの領域を単体的複体で表現するためにはそれぞれを単体のつながり方の条件を満たすように分割すればよい．ここでは三角面 Obc の動きによって定義される三角柱 Obcfed は3個の三角錐 Obcf，becf，edcf に分割される．同時に四角錐 Obefa は2個の三角錐 Obfa と三角錐 bfea の2個に分割する．その結果として得られる5個の三角錐の集合によって，2個の三角面によって定義される四角面が三角面になるプロセスをモデル化できる．

単体的複体 K は単体の有限集合であり，その元を $\sigma_1, \sigma_2, \ldots, \sigma_r$ とする．このとき，R^N の中で，$|K| = \sigma_1 \cup \sigma_2 \cup \ldots \cup \sigma_r$ によって図形 $|K|$ を定義する．これを単体的複体 K の多面体という．逆に，位相空間 $|K|$ は単体的複体 K によって複数の単体に分割され，組合せ的な構造を与えられる．言い換えると，単体の組合せ構造を持ったものが単体的複体 K であり，そのような構造を取り払って全体を1つの点集合と見なしたものが多面体 $|K|$ である．ある多面体に与えられる組合せ構造は一意には決まらないが，元となっている単体すべてを点集合とみて和集合をとれば，その和集合は分割の種類にはよらない．

多面体だけを扱うとすると，位相空間としてたとえば球面や球体といった基本的な図形を取り扱うことができない．そのため，多面体に対して同位相写像をうまく選んで位相空間を単体分割する．この単体分割を使えば，位相空間に関する位相的な特徴量は，単体分割されてできた単体的複体に関する位相的特徴量として計算できる．それゆえ，位相的な特徴量を計算するには多面体だけを考え，それに対応する単体的複体を取り扱うことができれば必要十分である．

また，位相空間がユークリッド空間のとき，単体的複体によって表現された多面体を使えば位相空間に関する計量的な特徴量の近似値を計算できる．これは位相空間の種類を問わず普遍的に適用できる方法である．使用する単体の数を増やしてゆけばいくらでも近似の精度を上げることができる．そのため，位相空間とし

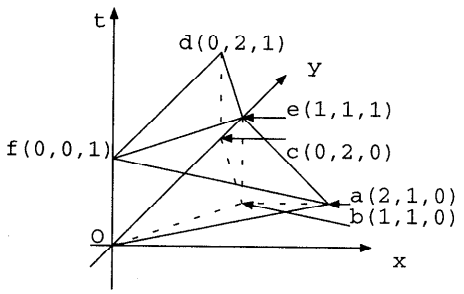

図5 凸胞複体の例
Fig. 5 Cell complexes.

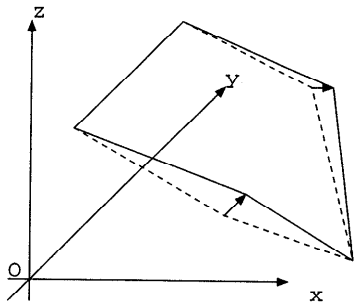

図6 位相的な情報と計量的なデータが食い違う例
Fig. 6 Inconsistency between topological and metric data.

て多面体を考えてそれを単体的複体を用いて表現すれば，計量的な特徴量の計算には実用的に十分である．そのため，位相空間データモデル Universe では，位相空間として多面体だけを考える．

2.3 単体的複体と凸胞複体の比較

多面体 $|K|$ を表現する方法として，単体的複体を用いる方法を説明したが，これ以外には凸胞複体（cell complex）[10]を用いる方法がある．いま，十分多数の半閉空間の積集合を凸胞体（cell）と呼ぶ．この凸胞体に対しても，単体に対して辺単体を定義したように辺凸胞体を定義できる．つまり辺凸胞体は凸胞体を定義する閉半空間の境界上の点全体が作る集合である．

このとき，凸胞複体（**図5**）とは有限個の凸胞体の集合 K であり，次の条件を満たす．

$\sigma, \tau \in K \Rightarrow$

$\sigma \cap \tau$ は空集合，または σ と τ の両方の辺凸胞体

図3にでてくる，四角面が三角面に変形するプロセスを凸胞複体で表現した例を図5に示す．三角面 Obc の動きによって定義される三角柱と，三角面 Oab の動きによって定義される四角錐はいずれも凸多面体であるから，これら2つの凸多面体を凸胞として表現し和集合をとって凸胞複体を定義すればよい．単体的複体表現との差は，図4と図5の点線（切れ目）に現れる．

単体的複体表現と，凸胞複体表現のいずれによっても，多面体を表現することができる．その際の利害得失は以下のようになる．

2.3.1 データ量

多面体を表現する場合，単体的複体を用いても凸胞複体を用いても多面体の頂点数に変わりはない．それゆえ，多面体の位相を記述するデータ量を比較することによって計算機内部での多面体の表現に必要なデータ量の大小を比較する．単体も凸胞体もどちらも凸集合であるから，辺の接続関係は一意に決まるため理論的には頂点集合で表現することができる．そうすると多面体を単体または凸胞体に分解したときの要素数によってデータ量はほぼ決まるが，これは凸胞体を用いた方が要素数の少ない分解が得られる場合が多い．そのためデータ量の面では凸胞複体を用いた表現の方が有利である．

2.3.2 操作の複雑さ

多面体に関する操作として，たとえば境界をとる演算を考える．この演算は多面体の境界を構成する辺単体または辺凸胞体の集合を求めることに等しい．これは多面体を構成する単体や凸胞体に向きを与え，それぞれの要素の辺をとり，それらに重みをつけて集計したものとして与えられる．しかし，このとき重みを計算するのは単体を用いた方が簡単であり，またアルゴリズムも容易である．

2.3.3 図形に関する性質

単体と凸胞体はいずれも凸図形であるという性質を持っている．単体の場合，この性質は実際的な局面，つまり浮動小数点表示された数で座標が表現された場合でも，任意の2頂点が等しくなければつねに成り立つ．また，単体は辺単体を構成するための最小数の点で定義されるため，同一の辺単体に含まれる頂点は必ず同一平面上にある．一方，凸胞体の場合は頂点位置の表現誤差によって凸胞体は必ずしも凸でなくなることがある．同様の理由から，凸胞体に含まれる辺凸胞体の頂点は理想的な条件下では同一の面上にあるが，浮動小数点数で表現された場合同一の面上にあることは必ずしも保証されない．たとえば，**図6**では本来凸多角形であるべき点線で表示された5角形は頂点座標の表現誤差によって凸でもなければ（下側の点線が実線に移る），同一平面にも存在しなくなる（上側の点線が実線に移る）．このように凸胞複体を用いて多面体を表現すると位相的情報と計量的な情報（凸性など）の食い違いが起こる可能性があるが，単体的複体によって多面体を表現すれば，位相的情報と計量的な

表1 集合論的演算子と位相述語の一覧．位相空間は単体的複体によって表現されている．

Table 1 List of operators provided by the topological space data model Universe.

演算子	入力 (1)	入力 (2)	出力
Intersection	位相空間	位相空間	位相空間
Union	位相空間	位相空間	位相空間
Difference	位相空間	位相空間	位相空間
Section	位相空間	位相空間	位相空間
Projection	位相空間	位相空間	位相空間
intersect	位相空間	位相空間	真または偽
contain	位相空間	位相空間	真または偽
meet	位相空間	位相空間	真または偽
disjoint	位相空間	位相空間	真または偽
equal	位相空間	位相空間	真または偽

情報（凸性など）の食い違いは起こらない．

2.3.4 Universeで用いた表現

以上の考察から，計算機内部でのデータ表現に要する記憶領域の観点からは凸胞複体を用いるのが便利である．一方，多面体に関する操作の単純さと現実の計算機環境での表現の安全性からは単体的複体を用いるのが良いと分かった．多面体に対する操作が単純であることと多面体を表現するときに位相的なデータと計量データの間に矛盾が起こることを回避できるため，データモデルUniverseにおいては単体的複体を用いて多面体を表現することにした．

2.4 演 算 子

ここでは位相空間に関する演算子として，このデータモデルUniverseの提供するもの（**表1**）を説明する．同時に，空間データや時空間データ，時間データに関する検索に演算子がどのように使われるかを説明する．その際，位相空間としては多面体を考える．それと同時に，その多面体は単体的複体によって表現されるとともに，その表現に基づいた組合せ構造が与えられている．

表1は図形の集合論的あるいは位相的な演算子のうちの基本的なものの一覧である．集合論的な演算としては積，和，差の3種類があり，それらを表す演算子を定義した．これらに関連して，ある面による図形の断面をとる演算子Sectionと，射影をとるProjection演算子を定義した．Section演算子はある時刻での図形の位置を計算するのに用い，Projection演算子はたとえば図形のsweepした空間や図形の存在時間を求めるのに用いる．また，図形の位相的な関係としては，交わるかそれとも離れているか，含むのか含まれるのか，等しいのかどうか，接するのかどうかという関係がある．これらの関係を表現するためには表1に示した演算子で十分と思われる．

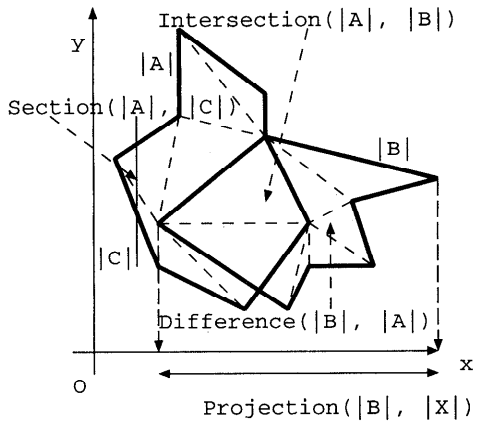

図7 集合演算子

Fig. 7 Set operators.

もちろんアプリケーションの種類によって必要とされる演算子の種類は異なる．たとえば距離関係を計算するための演算子や，方位に関する演算子，時空間的な関係に関する演算子を利用するためには，先の表に示した以外の演算子を定義する必要がある．この章と次章で，先の表に示されなかった演算子のいくつかをどのように定義するかを説明する．

2.4.1 集合演算子

集合演算子は，位相空間の和（union），差（difference），積（intersection）（**図7**）を求めるものである．これらの演算子は，たとえば，空間データベースにおいてはしばしば使用され，その意味や使用法はすでにあきらかになっている[6]．たとえば，2つの空間データの積はそれらの重なり部分を表している．そこで，ここでは時空間データにこの演算子を適用するとどのようなことができるのか説明する．

まず，2つの位相空間 $|A|$ と $|B|$ の和集合 $Union(|A|,|B|)$ を考える．位相空間を表現する単体的複体は有限個の単体の和集合であるから，2つの位相空間 $|A|$ と $|B|$ が共通部分を持たなければ，2つの単体的複体を構成する単体の集合もまた単体的複体となる．その結果できあがった新たな単体的複体の表現する位相空間は，2つの別々の位相空間が併合されてできた位相空間を表す．その意味で，この演算子は2つの時空間データの併合を決める時空間演算子 $Merge(|A|,|B|)$[8]を抽象化している．2つの位相空間 $|A|$ と $|B|$ が共通部分を持つとすると，和集合演算によってその2つの時空間データは互いに『食い込むようにして』併合されることを意味する．

次に，2つの位相空間 $|A|$ と $|B|$ の差集合 $Difference(|A|,|B|)$ を考える．この演算子は先の和

集合演算子の逆であると考えられる．この場合，時空間データに関する差集合演算子は，時空間データ $|A|$ から時空間データ $|B|$ が脱落するプロセスをモデル化すると同時に，時空間に関する分裂演算子 $Split(|A|,|B|)$ [8] を抽象化している．2つの時空間データが共通部分を持つとすると，その2つの時空間データは互いに『引きちぎれるようにして』分裂してできた一方の位相空間を意味する．

最後に，2つの位相空間 $|A|$ と $|B|$ の積集合 $Intersection(|A|,|B|)$ を考える．この演算子は先の差集合と組になっており，$Difference(|A|,|B|)$ によって時空間データ $|A|$ から脱落した時空間データの片割れを与える．

積集合演算子には断面演算子という変種がある．断面演算子 $Section(|A|,|B|)$ とは，面 $|B|$ による $|A|$ の断面を求めるものである．断面演算子が時空間データに適用されたとき，この面 $|B|$ がたとえば等時刻面であれば，その時刻における時空間データの位置と形が返される．また，その面 $|B|$ が空間的な制約だけによって決まるものである場合，位相空間 $|A|$ を $|B|$ に制限して得られる領域の時間変化が得られる．たとえば，地形データの時間変化 $|A|$ と標高 100 メートルを表す面 $|B|$ が与えられると，$|B|$ による $|A|$ の断面 $Section(|A|,|B|)$ は標高 100 メートルの地域の時間変化を表している．このように，断面演算子によってある制限された領域でのデータの時間変化が得られる．

射影演算子 $Projection(|A|,|B|)$ は，位相空間 $|A|$ を部分空間 $|B|$ に射影するものである．たとえば，$|B|$ が時間軸とすると，射影演算子 $Projection(|A|,|B|)$ は，位相空間 $|A|$ の存在する時区間を与える．また，$|B|$ が等時刻面であるとすると，射影演算子 $Projection(|A|,|B|)$ は，位相空間 $|A|$ として表現された時空間データが通過した領域を与える．たとえば，位相空間 $|A|$ が降水量が 1 mm 以上の領域の時間変化を表すとすると，射影演算子 $Projection(|A|,|B|)$ によって降水量が 1 mm 以上の領域の全体が得られる．

2.4.2 位相演算子

2つの位相空間に関する位相的な関係を記述するための演算子を位相演算子という．現在このモデル Universe では，位相演算子として位相述語が定義されている．位相述語とは，その述語が規定する2つの位相空間の位相的な関係，たとえば共通部分を持つかどうかという関係が成り立つかどうかを判定し，成り立っていれば真を，そうでなければ偽を返す演算子である．2つの位相空間に関する位相述語は $intersect(|A|,|B|)$（2つの位相空間が共通部分を持つときにだけ真），$disjoint(|A|,|B|)$（2つの位相空間が共通部分を持たないときにだけ真）の2つに分けられる．位相述語 $intersect(|A|,|B|)$ の変種として $contain(|A|,|B|)$ や，$meet(|A|,|B|)$ が与えられる．集合演算子のときと同様，ここでも時空間データに関して，その意味と利用法を説明する．

まず，$disjoint(|A|,|B|)$ という述語が真となる場合を考える．そのような場合は，2つの位相空間 $|A|$, $|B|$ が共通部分を持たないときであるが，これは2つの時空間データがそれぞれの生存期間のすべての時刻にわたって空間的に接点を持たないことを意味する．この述語はたとえば，1回も衝突しない（すべての時刻にわたって接点を持たない）時空間データの組を求めるのに使用する．たとえば，分子運動をモデル化したとき，衝突しなかった分子の組を取り出そうとするときに，この時空間述語を利用すれば検索条件を指定できる．

また，時空間述語 $intersect(|A|,|B|)$ を考える．これは2つの時空間データがそれぞれの生存期間の共通部分を表す時区間のどこかの時点で交わることがあれば真，そうでなければ偽となる述語である．この時空間述語は先の時空間述語 $disjoint(|A|,|B|)$ と時空間の意味でも否定になっている．この時空間述語を用いるとロボティックスやコンピュータアニメーションでの衝突検出の条件を，$intersect(|A|,|B|)$ と記述できる．

時空間述語 $contain(|A|,|B|)$ や，$meet(|A|,|B|)$, $equal(|A|,|B|)$ も上の2つの時空間述語と同様に解釈することができる．時空間述語の $contain(|A|,|B|)$ と $equal(|A|,|B|)$ は2つの時空間データの生存する期間のすべての時点にわたって，同名の空間述語 $contain(|A|,|B|)$ と $equal(|A|,|B|)$ が成り立つことと解釈される．また，時空間述語 $meet(|A|,|B|)$ は，それぞれの生存期間の共通期間のある時刻に同名の空間述語が成立することと解釈される．

また，時空間述語 $equal(|A|,|B|)$ については，2つの位相空間 $|A|$ と $|B|$ が点集合として等しいかどうかを調べる述語と，多面体を単体分割した結果得られる単体的複体表現の一致を問題にする述語を定義することができる．しかし，通常は点集合として2つの多面体が一致するかどうかを判定する述語 $equal(|A|,|B|)$ を用いれば十分である．というのも，多面体は点集合であり，その多面体の形状は単体分割の結果には依存しないからである．

これ以外の時空間的な位相述語も定義することができる．そのような述語は集合演算子と位相演算子を用

いて定義できる．たとえば，2つの時空間データを表す多面体であって，ある時刻において空間的に分離しているものを検索するとき，検索条件を指定する述語 $disjoint(|A|,|B|)\_for\_sometime$[17]を定義することも可能である．この述語はたとえば，

$disjoint(|A|,|B|)\_for\_sometime$
$= disjoint(|A|,|B|)$
$\vee equal(Projection(|A|,|T|),$
$Projection(Intersection(|A|,|B|),|T|)) = false$
$\wedge equal(Projection(|B|,|T|),$
$Projection(Intersection(|A|,|B|),|T|)) = false$

と展開できるので，これに従って新たに定義すればよい．

2.4.3 距離演算子

この演算子は，空間データや時間データを表す位相空間に適用できる．適用を受けるデータが空間データであれば，たとえばユークリッド距離に基づいて計算が行われる．時間データについても時区間の長さを用いて演算を行うことができる．たとえば，空間データ $|A|$ に対して空間データ $|B|$ がある距離 d 以内にあるかどうかという述語 $within\_distance(d,|A|,|B|)$ は，たとえば2つの多面体の間の距離をはかることで真偽判定される．

時空間データを表現する位相空間 $|A|$ と $|B|$ についても，それぞれに対して断面演算子を用いてある特定の時刻における空間データを取り出したり，射影演算子を用いて位相空間の生存時区間を取り出すことができれば比較を行うことができる．たとえば，ある時空間データ $|A|$ と同時期に存在する時空間データ $|B|$ を取り出すためには，時空間述語と同名の時間述語 $intersect()$ を用いて，$intersect(Projection(|A|,|T|), Projection(|B|,|T|))$ が真となる時空間データを取り出せばよい（ただし，$|T|$ は時間軸を表す位相空間）．また，生存期間が与えられた期間 $Interval$ より長いもの $|A|$ を取り出すときには，時間データに関して時区間の長さを取り出す演算子 $duration()$ を用いて，$duration(Projection(|A|,|T|)) \geq Interval$ が真になるような時空間データ $|A|$ を取り出せばよい．

また，ある指定された時刻に，ある時空間データから距離 d 以内にある時空間データを取り出すには，$within\_distance(d, Section(|A|,|H|), Section(|B|,|H|))$ という条件が真となる時空間データを取り出せばよい（ただし，$|H|$ は指定された時刻の等時刻面）．

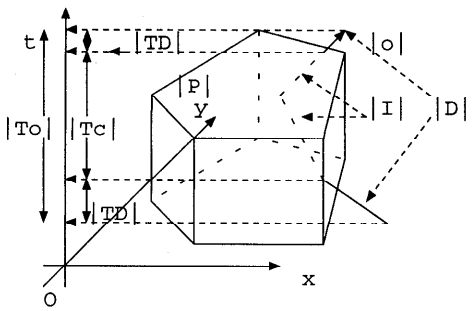

図8 オブジェクトoと多角形Pの位相空間における関係
Fig. 8 Relationship between an object o and a polygon P.

3. 時空間データ検索への応用

位相空間データモデル Universe を用いて表現された時空間データに対して，どのように時空間検索を記述するかを述べる．ここでは，Sistla ら[17]の時空間問合せの例題を用いて時間，空間，時空間条件の部分を抽出し説明する☆．

まず最初の問合せは，「ある将来の状態において，多角形Pの内側に入るすべてのオブジェクトo（the object that is contained in the polygon P eventually）」を求める問合せである．

ここで述べられた問合せは，多角形Pで表現された都市の領域に，車oが出入りするプロセスをモデル化したデータベースに対して，1時間後に市内に入っている車を調べる問合せを抽象化している．

検索されるオブジェクトoは，「ある将来の時刻にoがPに含まれる」が成り立てばよい．議論を単純化するために，すべてのデータが基準となる時刻以降に存在するように定義されていて，将来にしかデータは存在しないと仮定する．そのとき，oとPの関係を位相空間で表現すると図8のようになる．

上記の検索条件が成り立つためにはoとPの時間変化を表す位相空間 $|o|$ と $|P|$ が空でない共通部分 $|I| = Intersection(|o|,|P|)$ を持つことが必要である．そのとき，共通部分 $|I|$ はoとPの共通部分の時間変化を記述していて，ある時刻 t_1 でoがPに含まれるとすると，時刻 $t = t_1$ を表す等時刻面による $|I|$ と $|o|$ の断面は一致する．よって，$|o|$ から $|I|$ を取り去ることによって得られる差集合 $|D| = Difference(|o|,|P|)$ を作ると，空間的にoがPに含まれている時区間では $|D|$ は点を持たない．そのため $|D|$ と $|o|$ を時間軸に射影することによっ

☆ 簡単のため，時間，空間，時空間に関係しない条件は省略した．

て得られる時区間 $|T_D| = Projection(|D|, |T|)$ と $|T_o| = Projection(|o|, |T|)$ は異なり，前者は後者の真部分集合になっている（$|T_D| \subset |T_o|$）．これはすなわち時区間 $|T_c| = Difference(|T_o|, |T_D|)$ にわたってオブジェクト o が多角形 P に含まれることになる．そこで，位相空間 $|P|$ が位相空間 $|o|$ をいつかは含むときにだけ真となる上位の位相述語 $contain\_eventually(|P|, |o|)$ を，

$contain\_eventually(|P|, |o|)$
$= intersect(|P|, |o|)$
$\wedge |T_D| = Projection(Difference(|o|, |P|), |T|)$
$\wedge |T_o| = Projection(|o|, |T|)$
$\wedge Equal(|T_D||T_o|) =$ "$False$" と定義する．そうするとこの問合せは以下のように記述される．

Select o
Where $contain\_eventually(|P|, |o|)$

2 番目の問合せは，「1 番目の問合せの条件を満たすもののうちで，P に含まれる時間が 2 単位時間であるオブジェクト o」をすべて求めるものである．この問合せで追加された条件である「P に含まれる時間が 2 単位時間である」という条件は，前のパラグラフで定義した時区間 $|T_c|$ の長さが 2 単位時間であるということになる．そのため時間データに固有の演算子 $duration()$ を与えられた時間データの長さを与える演算子として定義すると同時に，位相空間 $|o|$ が位相空間 $|P|$ に含まれる時区間を与える上位の演算子 $duration\_for\_contain(|P|, |o|)$ を

$duration\_for\_contain(|P|, |o|)$
$= duration(Difference(|T_o|, |T_D|))$

ただし，

$|T_o| = Projection(|o|, |T|)$
$|T_D| = Projection(Difference(|o|, |P|), |T|)$

と定義する．これを用いて「P に含まれる時間が 2 単位時間である」という条件は

$duration\_for\_contain(|P|, |o|) =$ "2"

と表現される．そのため，2 番目の問合せは次のように記述される．

Select o
Where $contain\_eventually(|o|, |P|)$
$\wedge duration\_for\_contain(|P|, |o|) =$ "2"

3 つめの問合せは，2 つめの問合せに加えて「多角形 P に含まれなくなった時間から 5 単位時間以内に別の多角形 Q にオブジェクト o が含まれる」という条件が追加されたものである．このとき多角形 Q の時間変化を表す多面体を $|Q|$ と定義する．あらたに加わった条件のうち「別の多角形 Q にオブジェクト o が含まれる」という部分は 1 つめの問合せの条件と同じである．そこで「多角形 P に含まれなくなった時間から 5 単位時間以内に」という部分をどのように書き下すかが問題となる．

これを書き下すためには，オブジェクト o が多角形 P，Q に含まれている時区間を求めてそれらの時間的な関係を調べればよい．オブジェクト o が多角形 Q に含まれる時区間 $|T_{cQ}|$ は上と同様に，

$|T_{cQ}| = Projection(Intersection(|o|, |Q|), |T|)$

と書ける．ここで時区間に対してその最大値と最小値を与えるような時間データに関する上位の演算子 $max\_of\_duration()$ と $min\_of\_duration()$ を定義すると，「多角形 P に含まれなくなった時間から 5 単位時間以内に別の多角形 Q にオブジェクト o が含まれる」という条件は，

$contain\_eventually(|Q|, |o|)$
$\wedge min\_of\_duration(|T_{cQ}|)$
$- max\_of\_duration(|T_o|)$
$\leq$ "5"

となる．よってこの問合せは以下のように記述される．

Select o
Where $contain\_eventually(|o|, |P|)$
$\wedge duration\_for\_contain(|P|, |o|) =$ "2"
$\wedge contain\_eventually(|Q|, |o|)$
$\wedge min\_of\_duration(|T_{cQ}|)$
$- max\_of\_duration(|T_o|)$
$\leq$ "5"

Sistla らの例題は上記の 3 つであるが，これらはいずれも Universe データモデルの演算子を用いて記述されることが分かった．それらの例題以外にも，時間，時空間問合せの例題[21]を，Universe データモデルの演算子を用いて記述できることを確認した．

4. 他のデータモデルとの比較

位相空間データモデルは空間データモデルや時間データモデル，時空間データモデルのいずれにも適用可能である．ここでは，これまでに提案されてきた時空間データモデルとの比較を試みるが，それに際して代表的なものをいくつかとりあげる．1 つは Space-Time Approach に基づいたものである．また，スナップショットビュー（snapshot view）に基づくものもあるのでそれとも比較を試みる．

4.1 時空間データモデルとの比較

4.1.1 Space-Time Approach に基づくモデル

2 章で説明したように，空間と時間の直積空間を定義し，その空間の図形として空間データの時間的変化

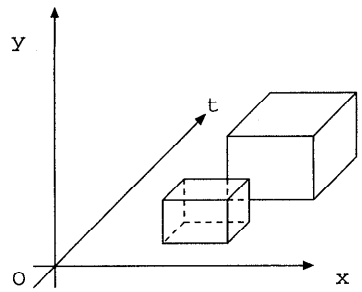

図9 Aritsugiらのアプローチによる位相空間表現
Fig. 9 Representation of topological spaces by Aritsugi's approach.

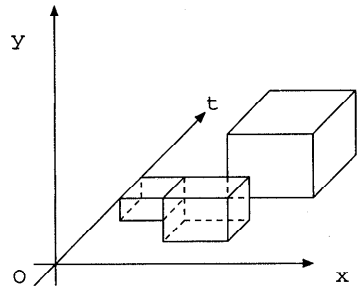

図10 Masunagaのアプローチによる位相空間表現
Fig. 10 Representation of topological spaces by Masunaga's approach.

を表現するのがSpace-Timeアプローチである．このアプローチを用いているものにはたとえば，Aritsugiら[2]，Bimboら[3]，Masunaga[14]，Worboys[22]がある．これらのアプローチにおいては，位相空間はどれも多面体を用いて表現されている．

Aritsugiらのアプローチにおいては，位相空間はMBB（Minimum Bounding Box）の集合によって表現されている（図9）．MBBとは，境界を形成する辺が座標軸と平行または直交する長方形や直方体である．このアプローチにおいては，あらゆる時区間において空間データは不変であり，1つの長方形や直方体で表現される．一方，区間の端点で空間データは不連続に変化する．そのため2つの時区間の境界となっている時点で，そこで接しているMBBどうしは単体的複体や凸胞複体のような接続の条件を満たさない．また，このアプローチでは，位相空間はMBBという特殊な凸胞体だけを用いるために，連続的に変化する空間データをモデリングすることができない．また，位相空間の位相を現実の計算環境においては完全に，また無矛盾に記述することが我々の方法に比べて困難である．

その一方で，モデルが簡単であり，またすべての面が座標軸に平行または直交するという条件から，時空間的な演算子を容易に計算することができる．また，時空間データから時間的あるいは空間的なデータを取り出すことも容易である．それゆえ，時間的なあるいは空間的な条件を元に検索することは容易である．しかし，MBBの集合を用いて位相空間を表現すると，その表現は我々の表現よりも不正確であるから検索結果にまぎれこむ『にせの解』（ここでは，MBBによる表現では検索条件を満たすが，位相空間自体は条件を満たさないもの）の数は多くなる．

Bimboらのアプローチでも，位相空間を1つのMBBで表現している．これによって表現精度は最も悪くなるが，データ表現が最も単純であるために検索に要する計算量は最も少ない．

Masunagaのアプローチにおいては，Aritsugiらのアプローチと同様にMBBの集合によって多面体を定義している（図10）．このアプローチとAritsugiらのアプローチの違いは，ある時区間にわたる空間データの表現にある．Masunagaのアプローチにおいては，ある時区間での空間データは複数のMBBの集合で表現されるが，Aritsugiらのアプローチでは1つのMBBで表現されていることである．また，Masunagaのアプローチでは空のMBBを導入しようと試みている．これにより位相空間の表現の精度と効率は向上するが，依然としてAritsugiらのアプローチ同様連続的に変化する空間データを表現することは困難であり，また位相空間の位相を現実的な計算機環境では無矛盾に表現できない．また，表現精度と効率をあげるためにAritsugiらのアプローチよりも多数のMBBで位相空間を表現したために，検索条件を満たすかどうかの計算量はAritsugiらのアプローチに比べて増大している．

Worboysのアプローチでは，時空間データの初期状態（空間データとして表現する）を単体的複体で表現する．単体的複体として表現された空間データに対して時区間の直積を定義する．たとえば2次元の空間データが単体的複体として表現されたとき，その時間変化を積複体[10]と呼ばれる特殊な凸胞体を用いて表現している（図11）．

たとえば，2次元空間にある点が等速直線運動しているとき，その点の動きを表す積複体は直線となる．また，2次元空間にある線分が等速直線運動しているとき，その線分の動きを表す積複体は四角形となる．同様に，三角形の動きを表す積複体は三角柱となる．

このように積複体を用いることによって剛体の並進運動を効率良く，また精度良く表現することができる．

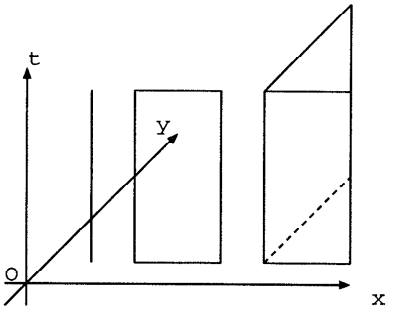

図 11 Worboys のアプローチによる位相空間表現
Fig. 11 Representation of topological spaces by Worboys's approach.

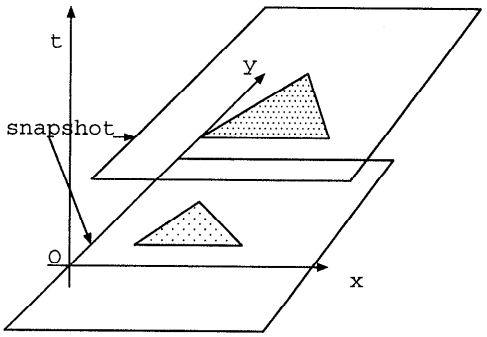

図 12 スナップショットビューによる表現
Fig. 12 Representation of spatio-temporal data by snapshot view approach.

しかしながら，積複体を用いることによって弾性体の動きをうまく表現することができない．なぜなら積複体によっては空間データのトポロジーの変化を記述することができないからである．その意味で我々のモデルは Worboys のアプローチよりもより多くの対象を表現できる．ただし，Worboys の表現できる対象に関しては，その表現に要するデータ量は多くなる．

4.1.2 スナップショットビューに基づくモデル

Kämpke のアプローチ[9]と Raafat らのアプローチ[16]はともに，スナップショットビューを用いたモデルを用いている．これはあたかも写真をとるようにある瞬間のデータを記録する手法である．記録法から明らかなように，この手法は離散時間モデルの典型である．この方法においては，同一スナップショット内のすべてのデータは同一の時間を持つので，個別のデータに関して時間の管理は必要ない．このモデルを用いた表現は，同一スナップショット内のデータを等時刻面の位相空間として表現することに等しい（**図 12**）．たとえば，2 次元空間内を移動する 2 次元図形のスナップショットは，3 次元位相空間内の 2 次元図形として表現される．しかし，Space-TimeApproach を用いて Universe データモデルで表現すると，これは 3 次元の位相空間となる．この違いのために，Universe データモデルではスナップショットをとった時刻以外のデータに関しても検索を行うことができる．

スナップショットビューに基づいて時空間データを表現した場合，同一の図形の各時刻での位置関係を記述しない．そのためスナップショットビューモデルで仮に各時刻での図形の表現に単体的複体を用いた場合，時間軸にそって稜線や面が発生せず，その分データ量は Universe データモデルよりも少ない．

時空間的な演算を行う際には，各時刻での図形の空間関係を計算し，その結果をもとにして最終的な演算結果を計算する必要がある．たとえば，2 つの図形の動きをスナップショットビューモデルで表現したとき，それら 2 つの図形の衝突を検出するには各時刻でのそれぞれの図形の位置から述語 intersect が真かどうかを判定し，それを総合して時空間的な述語の真偽を計算する．一方，Universe データモデルでは等時刻面間の図形の動きを記述している．そのため Universe データモデルではより次元の高い図形を，またより多くの図形を対象として演算を行うため，スナップショットビューモデルに比べて演算時間が長くなる．

4.2 空間データモデルとの比較

位相空間データモデルは，空間データモデルとしても利用可能である．そこで空間データモデルとして用いられるいくつかのモデルと比較する．

4.2.1 ラスタ型データ表現

ラスタ型データ表現[1]は地図や画像のモデルとして用いられる表現法である．地図の場合，たとえば，地表面を等間隔の格子に区分し，各格子点におけるデータを格子点に与えられた順序に従って並べたものである．それゆえ，各格子点の座標とデータの直積を作ることによって位相空間における点集合データとして実現できる．その結果，我々のモデルでこの表現法をシミュレートできる．

ラスター型データ表現を用いて空間データを表現する場合，図形の内部に含まれる格子点の数に等しい頂点データが必要である．一方，単体的複体を用いて図形データを表現した場合，境界に含まれる頂点データの座標の集合と，図形が何個の単体の集合で表現されるかによって決まる．たとえば，凸 n 角形領域を単体的複体で表現すると，必要となる頂点が n 個，単体数が $n-2$ 個であるから，頂点数に比例している．一方，ラスター型表現を用いると表現に必要な頂点数は一般に n よりも大きい（これは領域の面積が大きければ大きいほど顕著になる）ので，ラスター型デー

タ表現の方が必要とするデータ量は大きい．

4.2.2 ベクター型データ表現

ベクター型データ表現[1]は，座標を用いて表現された点データを用いる．そして点データを辺で結んで面の境界を定義し，さらに面の集合によって領域を表現する．面をつねに凸多角形にするとこのように表現されたデータは凸胞複体となる．また面がつねに三角形であれば，単体的複体となる．また，面が凸でない場合には境界を構成する辺の順序を正しく記録する必要がある．面が凸でない場合には適切に仮線をいれて面が三角形になるように分割することによって単体的複体に変換することができる．したがって，ラスタ型データ表現と同様に，Universeデータモデルによってベクター型データ表現をシミュレートできる．

同一の2次元空間の多角形をベクター型およびUniverseデータ表現を用いて表現した場合を考える．このとき，この多角形が凸 n 多角形である場合を考える．頂点の位置を表す座標データは共通であるから，点を結ぶ稜線の集合の表現が問題になる．ベクター型表現では，稜線が n 本であることから，稜線の両端点の名前の対を n 組用意すればよい．それゆえ $2n$ 個の頂点名で記述できる．一方，Universeデータ表現では，凸 n 角形は少なくとも $n-2$ 個の三角形（2-単体）に分解され，それぞれの三角形は3個の頂点集合によって記述されるから，少なくとも $3(n-2)$ 個の頂点名で記述される．凸でない多角形は凸多角形の集合に分解されて表現されるが，それぞれの凸多角形の表現に関してUniverseデータ表現の方がデータ量は多くなるため，全体としてUniverseデータ表現の方が必要なデータ量は多い．

一方，集合演算や位相述語を計算するとき，図形に含まれる各格子点の名前の集合の一致を計算することによって演算を行うことができる．そのため，2つの図形を表す格子点の数の積に比例する．同じ図形を単体的複体によって多角形として表したとき，演算の手間はそれぞれの多角形を構成する三角形の数の積によって表現される．そのため，図形の大きさが小さく，それぞれの図形の三角形の数と格子点の数が同程度であれば演算の手間は同程度である．また，図形の面積が大きくなって格子点数が三角形数よりもずっと大きくなると演算の手間はラスター方式の方が大きくなる．

4.2.3 境界表現

境界表現[20]は，CADやコンピュータアニメーションにおいて形状を記述するために用いられる手法である．境界表現は三次元領域を定義する際に，その領域の境界を構成する面の集合を用いる．また面を定義するために，その面の境界を定義する辺の集合を用いる．同様に辺はその両端点によって定義する．このように階層的に，ある領域をその境界で表現することによって定義する．単体的複体による多面体の表現は，多面体の境界表現とは異なる．しかし，単体的複体に対する境界演算子 δ を用いることによって多面体の境界を求めることができるので，単体的複体表現を境界表現に変換できる．逆に，境界表現された領域に対して，面を単体分割しそれをもとに領域を単体分割すれば，この分割の結果得られる領域表現は単体的複体となる．そのため，境界表現とUniverseデータモデルは，互いに変換可能である．

境界表現では凸多面体を，その境界を構成する面の集合で表現する．たとえば，凸多面体を単体的複体で表現し，その境界を構成する面（この場合は2-単体）を取り出すことでその多面体の境界表現を定義することができる．このとき境界を構成する面の数は単体的複体を構成する単体の数よりも少なく，また，境界を構成する面の次元は単体的複体の次元より1小さい．そのため，境界表現によって凸多面体を表現するほうがUniverseデータ表現を用いて表現するよりもデータ量は少ない．

それぞれの表現に基づいて演算を行う場合，たとえば凸多面体どうしの関係を計算する場合，図形の凸性を利用したアルゴリズムを用いることによって，単体的複体よりも効率良く計算を行える．一方，凸でない多面体（たとえば穴の空いているもの）の間の関係を計算するためには，境界表現の場合いったん凸胞分割を行って凸多面体どうしの関係計算に帰着させることが一般的である．ところが単体的複体による表現ではすでに凸図形である単体に分割されており，凸胞分割の前処理を必要としない．そのため単体的複体を用いて計算したほうが手間が少ない．

4.2.4 単体的複体による表現

単体的複体を用いて空間データをモデル化することがEgenhoferら[5]によって試みられた．彼らの取り扱う対象はおもに2次元空間データに限られていたが，我々は3次元空間データだけでなく時間データと時空間データに関しても単体的複体を用いる．

また，Egenhoferらは単体的複体を用いて表現された2次元空間データに対して，点，線分，多角形の挿入演算を定義した．これらの挿入演算はUniverseデータモデルでは，位相空間の和集合演算として，より一般的な形で定義されている．挿入演算以外にEgenhoferらは境界演算boundaryとその逆演算co-boundaryを定義している．boundary(C)とは，単体的複体Cが

n 次元であるとすると，C の表す多面体の境界を構成する $n-1$ 次元の辺単体の集合を求める演算子である．特に C が n-単体であれば，boundary(C) は C の $n+1$ 個の $(n-1)$-辺単体を与える．一方，co-boundary(S) とは，単体 S を引数とする演算子であり，単体 S を辺単体とする単体の集合を返す．これらの演算子を使えば，位相空間として表された空間データ，たとえば市町村の領域データが与えられたときに，その市町村境を boundary 演算子によって記述できる．また，境界線とっている単体を指定すれば，その単体を境界として接している市町村を検索できる．

boundary 演算子と co-boundary 演算子は Universe データモデルでは定義されていない．それは，Universe データモデルでは，位相空間の位相的関係の取扱いが最も大きな目的であり，位相的な関係を記述する演算をまず最初に定義したからである．境界演算は Universe データモデルで定義可能であり，また単体的複体に関するホモロジー[10]の計算法に従って計算できる．これら演算子の実装は今後の課題である．

また，データモデル Universe においては位相空間に関する 2 項演算子として集合演算子や位相演算子と定義している．これらの演算子を時空間データに適用したときの意味について我々は論じている．しかし，これら 2 項演算子のうち，和集合演算以外は Egenhofer らは定義していない．

4.3 時間データモデルとの比較

時間データモデル[19]においては，ある時区間にわたって一定の値をとるデータの表現が研究された．たとえば，従業員の給与の時間変化を記述するとき，給与はある一定期間変動しないと仮定する．このとき時間データモデルでは，給与 salary は時区間の開始時刻 start と終了時刻 stop が添付され 3 つ組 (salary, start, stop) で表現される．一方，Universe データモデルにおいては，2 点 (salary, start)，(salary, stop) を端点とする線分によって salary は 2 次元空間中の 1 次元位相空間として表現される．これは，時間問合せ言語 TSQL[18]での時間データの表現を幾何学的に表したものと一致する．

5. おわりに

単体的複体を用いた位相空間表現を用いた位相空間データモデル Universe を提案した．位相空間は図形の概念を一般化したもので，地図や CAD に現れる図形のみならず，非空間的なデータを図形化して得られる仮想的な図形についてもまったく同じ形で表現・格納することができる．そのためこのデータモデル Universe は幅広い適用領域を持っている．たとえば，時空間データを Space-Time Approach を用いて位相空間に変換することで，このモデル Universe で時空間データを取り扱うことができる．Space-Time Approach を用いて時空間データを位相空間表現することがこのモデルの特徴である．

単体的複体を用いて位相空間を表現することによって表現のためのデータ量は増える．その一方で単体的複体を用いて表現することにより，構成要素である単体はつねに凸であることが保証され，位相的な情報と計量的な情報の食い違いを減らすことができる．同じことは凸胞複体を用いても行うことができる．しかし，現実の計算機環境においては実数の浮動小数点表示のために凸胞体の凸性を保証することができないため，単体的複体を用いた．

また，単体的複体を用いて表現された位相空間に関する演算子を定義した．時空間データを位相空間として表現したときに，これら演算子の意味するところを説明すると同時にどのように使用するか述べた．

今後の課題としては，実際に位相空間データを作成し，位相空間を表現するのに必要な計算機内部でのデータ量を見積もると同時に，演算子の実行にかかる計算量を評価することがあげられる．また，我々はここで定義した演算子の実装を行っている[13]が，演算子の計算のために我々が設計・実装したアルゴリズムが破綻なく終了するかどうか確認する必要がある．また，このモデルを用いて実際にアプリケーションを構築することも課題である．

謝辞 この研究の一部は文部省平成 10 年度科学研究費補助金重点領域研究（課題番号 08244105）の補助を受けている．

参考文献

1) 秋山 実：地理情報の処理，山海堂 (1996).
2) Aritsugi, M., Tagashira, T., Amagasa, T. and Kanamori, Y.: An Approach to Spatio-Temporal Queries – Interval-Based Contents Representation of Images, *Proc. Int. Conf. Database and Expert Systems Applications*, pp.202–213 (1997).
3) Bimbo, A.D., Vicario, E. and Zingoni, D.: Symbolic Descrition and Visual Querying of Image Sequences Using Spatio-Temporal Logic, *IEEE Trans. Knowledge and Data Engineering*, Vol.7, No.4, pp.609–622 (1995).
4) Cameron, S.: Collision Detection by Four-Dimensional Intersection Testing, *IEEE Trans. Robotics and Automation*, Vol.6, No.3, pp.291–

302 (1990).
5) Egenhofer, M.J., Frank, A.U. and Jackson, J.P.: A Topological Data Model for Spatial Databases, *Proc. 1st Symp. Spatial Databases*, pp.271–286 (1989).
6) Egenhofer, M.J. and Franzosa, R.: On the Equivalence of Topological Relations, *Int. Journal of Geographical Information Systems*, Vol.9, No.2, pp.133–152 (1995).
7) Glassner, A.S.: Spacetime Ray Tracing for Animation, *IEEE Trans. Computer Graphics and Applications*, Vol.8, pp.60–70 (1988).
8) Hornsby, K. and Egenhofer, M.J.: Qualitative Representaion of Change, *Spatial Information Theory*, Hirtle, S. and Frank, A.U. (Eds.), pp.15–33, Springer (1997).
9) Kämpke, T.: Storing and Retrieving Changes in a Sequence of Polygons, *Int. Journal of Geographical Information Systems*, Vol.8, No.6, pp.493–513 (1994).
10) 河田敬義：位相幾何学，岩波書店 (1965).
11) 加藤俊一，栗田多喜夫：画像の内容検索 電子美術館への応用，情報処理，Vol.33, No.5, pp.466–477 (1992).
12) Kuroki, S., Ishizuka, K. and Makinouchi, A.: Towards a Spatio-Temporal OQL for the Four Dimensional Spatial Database System Hawks, *Proc.8th Int. Workshop on Database and Expert Systems Applications*, pp.142–147 (1997).
13) 黒木 進，尾下真樹，牧之内顕文：凸胞複体表現されたd-多面体の集合演算アルゴリズム，九州大学大学院システム情報科学研究科報告，Vol.4, No.1 (1999).
14) Masunaga, Y.: The Block-World Data Model for the Realization of Three-Dimensional Virtual Work Space, *Proc. Int. Symp. Digital Media Information Base*, pp.1–10 (1997).
15) Peuquet, D.J. and Duan, N.: An Event-Based Spatiotemporal Data Model (ESTDM) for Temporal Analysis of Geographical Data, *Int. Journal of Geographical Information Systems*, Vol.9, No.1, pp.7–24 (1995).
16) Raafat, H., Yang, Z. and Gauther, D.: Relational Spatial Topologies for Historical Geographical Information, *Int. Journal of Geographical Information Systems*, Vol.8, No.2, pp.163–173 (1994).
17) Sistla, A.P., Wolfson, O., Chamberlain, S. and Dao, S.: Modeling and Querying Moving Objects, *Proc. 13th Int. Conf. Data Engineering*, pp.422–432 (1997).
18) Snodgrass, R.T. (Ed.): *The TSQL2 Temporal Query Language*, Kluwer Academic (1995).
19) Tansel, A.U., Clifford, J., Gadia, S., Jajodia, S., Segev, A. and Snodgrass, R.: *Temporal Databases – Theory, Design, and Implementation*, Benjamin/Cummings (1993).
20) 鳥谷浩志，千代倉弘明（編）：3次元CADの基礎と応用，共立出版 (1991).
21) Tsotras, V., Jensen, C. and Snodgrass, R.: An Extensible Notation for Spatiotemporal Index Queries, *SIGMOD Record*, Vol.27, No.1, pp.47–53 (1998).
22) Worboys, M.: A Unified Model for Spatial and Temporal Information, *The Computer Journal*, Vol.37, No.1 (1994).

(平成10年5月20日受付)
(平成11年2月 8 日採録)

黒木 進（正会員）

昭和63年東京大学工学部計数工学科卒業．平成2年同大学院工学系研究科計数工学専攻修士課程修了．同年九州大学工学部情報工学科助手．平成8年より同大学院システム情報科学研究科助手．マルチメディアデータベース，時空間データベースの研究に従事．

牧之内顕文（正会員）

昭和42年京都大学工学部電子工学科卒業．昭和45年グルノーブル大学理学部応用数学科 Docteur-Ingènieur 取得．(株) 富士通，(株) 富士通研究所，九州大学工学部教授を経て，平成8年同大学院システム情報科学研究科教授．ACM，IEEE Computer Society 各会員．