5K-1

空間データベースシステム Hawk's Eye の実装と評価

金子 邦彦　牧之内顕文

九州大学 大学院 システム情報科学研究院　〒812-8581 福岡市東区箱崎 6-10-1

1. はじめに

種々の分野にアプリケーションを持つ空間データベースシステムの重要性は高い．我々は，３次元及びそれ以上の空間物を扱える空間データベースシステム Hawk's Eye の研究開発を行ってきた．本稿では，Hawk's Eye に実装された空間モデルである cell complex[1] について，そのデータサイズ，空間オペレーションの実装と評価について報告する．

空間データベースでの空間問い合わせ[2]には，種々の空間述語，空間オペレーションが登場する．いずれも，空間物が持つ各種の位相，幾何の情報を使う．空間述語は，指定された条件に応じて true, false 値を返し，空間オペレーションは，数値，空間物を結果として返す．これらは，オペランド数によって大別できる．詳しくは表１の通りである．空間データベースシステムでは，これらが全て高速に処理される必要があり，このことが重要な研究課題である．

我々が，cell complex を Hawk's Eye に実装した理由は以下の通りである．Cell complex は，互いに重ならない cell の集合として空間物を表現する． Cell Complex を，サーフェスモデルや，CAD などの特定分野で使われている種々のソリッドモデル[3]と比較すると，表現可能な空間物の種類（任意の次元，任意の種類の空間物を一様に表現できる），表現できる情報（幾何情報，cell 間の位相情報）で優れ，種々の空間モデルの中で最も一般性があり強力である．現在，cell complex は，種々のアプリケーションで利用されている．Cell complex は，cell を node とし，１つ次元が異なる 2 つの cell 間の boundary, co-boundary 関係を arc とするグラフである incidence graph [4] として表現されてきた．Incidence graph として表現された cell complex では，表１に示したようなオペランド数が１の空間述語，オペレーションは効率よく動く．以上の点で，incidence graph 表現された cell complex はメリットがある．オペランド数２のものは，5 章で議論する．

表１．空間述語，空間オペレーション

	オペランド数１	オペランド数２
空間述語	<u>位相</u> ・連結性 ・manifold か	<u>位相</u> ・intersect ・contain ・equal
空間オペレーション	<u>位相・幾何</u> ・boundary 取得 ・指定次元の要素の取得 <u>計量</u> ・体積，直径	<u>位相・幾何</u> ・intersection ・difference

2. Cell

以下，cell complex をΓ，cell をσ，d 次元の空間を $\mathbf{R}^d$，空間中の点を p，k 個の超平面集合を $\mathbf{H} = \{h_1, h_2, ..., h_k\}$，超平面 h_i の方程式を $\phi_i(p)$ と書く．

Cell は，点，線分，ポリゴンなどの凸空間物を，次元について一般化した概念である． Cell には，有界 cell，非有界 cell の 2 種がある，ある数値 r を定めたとき，cell σ 内の任意の 2 点 p_1, p_2 について，その距離 $dist(p_1, p_2)$ が常に r 以下になるとき，σ は有界 cell である．さもなければσは非有界 cell である．Cell は線形方程式，線形不等式の集まりを満足する点集合である[5]．線形方程式を超平面としてとらえ，線形不等式は超平面を境界としてもつ半空間としてとらえることができる．以上のことから，超平面集合 **H** を使って，有界，非有界 cell ともに，$\sigma = \{p \mid \phi_i(p)\ \boldsymbol{op}\ 0, \boldsymbol{op} \in \{\leqq, =, \geqq\}, p \in \mathbf{R}^d\}$ のように表現できる． Cell σ がこのように超平面集合 **H** で表現されているとき，我々は，**H** の各超平面を σ の relevant 超平面と呼ぶことにしている．

3. Cell Position Vector Structure

Incidence graph をコンピュータ上に表現する空間データ構造としては，incidence graph の arc をポインタとして表現する方法，incidence graph の top cell と 0-cell 間の path をタップル(tuple)として表現する cell-tuple 構造[6] が従来から知られている．３次元あるいはそれ以上の次元を持つ cell complex の incidence graph は，多数の node, arc, path を含むので，従来のデータ構造よりもコンパクトに表現できるようなデータ構造が必要と考え，cell position vector structure を考案した．

以下，cell complex Γ の top cell の relevant 超平面集合全ての和を $\mathbf{H}_{rel}$ (超平面数は k 個とする)と書き，Γ の relevant 超平面と呼ぶ． Γ 内の cell に

Implementation and Evaluation of a Spatial Database System named Hawk's Eye,
Kunihiko Kaneko and Akifumi Makinouchi
Graduate School of Information Science and Electrical Engineering, Kyushu University

対して，H_{rel} の超平面 h_j が relevant 超平面で無いとき u_j = i ，対応するオペレータ ***op*** = '=' のとき u_j = 0，***op*** = '≧' のとき u_j = +，***op*** = '≦' のとき u_j = - という値を持つようなベクトル **u** = [u_1, u_2, ..., u_k] を，σ の cell position vector と定義する．H_{rel} 全ての超平面の方程式と，σ の cell position vector **u** が与えられたとき，**u** を使って，H_{rel} から σ の relevant 超平面（u_j ≠i のもの）を選び，それを使って空間を分割し，分割してできる incidence graph から **u** を使って σ の外側を取り除くと，もとの cell σ の incidence graph が得られる．以上のことから，Γ の incidence graph は，H_{rel} とΓ の全ての top cell の cell position vector によって表現できる．これが，cell complex の Cell Position Vector Structure(**CPVS**) である．Cell Position Vector Structure には，多数の i が登場するので，簡単な方法で圧縮できる．

4. データサイズ評価

cell complex Γ のデータサイズを見積もるために，次のパラメータを定める．

N_0 : Γ の 0 次元 cell 数
N_{top} : Γ の top-cell 数
N_{cell} : Γ の 総 cell 数
N_{rel} : Γ の relevant 超平面数
α : Γ の top-cell の relevant 超平面数の平均
β : Γ の 0-cell 数の relevant 超平面数の平均

incidence graph には，最低でも d! N_0 個の path がある．cell-tuple では，１タップルのサイズは，1/8 × (d+1) × log N_f バイトと見積もられる(d+1 はタップルの長さ，log N_f は，タップルに格納される cell 番号の長さ)．これと別に空間物の形を記述するための頂点の座標データで 8N_0 バイトを要する．タップル数は d! N_0 個なので，データサイズは，1/8 × (d+1)! × N_0 × log N_f ＋8dN_0 である．一方 **CPVS** では，top cell の cell position vector では，現在，Hawk's Eye に実装している圧縮法では，1/8 ×α×N_{top} ×log N_{rel} バイトを必要とする．また，超平面の方程式を格納するために，現在の実装では，8dN_0＋1/8 ×β×N_0 ×log N_{rel} バイトを必要とする．比較のために，空間次元 d = 3 とし，N_0　N_{top}, α　d+1, β　d を仮定すると，

Cell Tuple: N_0 (3×log N_{cell}+ 8d)
CPVS: 　N_0 (0.875×log N_{rel}+ 8d)

が得られる．N_{cell} > N_{rel} なので，明らかに，CPVS の方がデータサイズは小さい．このことを実験的に裏付けるために，３次元空間中にランダムに発生させた超平面を使って，３次元の cell complex を人工的に生成し，cell complex を cell-tuple で表現したときと CPVS で表現したときの２手法のデータサイズを比較した．実験結果を表２に示している．実験では，N_{rel}=26, N_{top} = 40, 80, 120, 160, 200 に設定して，cell complex を発生させた．この実験結果も CPVS の方がデータサイズは小さいことを示している．

表２．３次元ランダム cell complex での cell tuple と CPVS のデータサイズ比較実験結果

N_{top}	CPVS [バイト]	cell-tuple [バイト]
40	7,537	19,632
80	13,985	35,320
120	18,273	48.168
160	22,426	57,216
200	25,037	59,952

5．空間述語，オペレーションのアルゴリズム

３次元以上でも動く空間述語のアルゴリズムとしては，線形計画法[5] あるいは双対空間にマッピングされた超平面を使って，空間物同士の intersection（交わっているかの判定）を求めるアルゴリズムはあった．intersection 以外の述語，オペレーション（表１参照）については，Arrangement Construction[7]を利用した方法が知られていた．我々は，cell complex の incidence graph を使って，２つの cell complex 同士の各種の空間述語，オペレーションを行うアルゴリズムを作った．これは，incidence graph を効果的に利用して，intersection（積），difference（差），contain（包含），equal（形状の同一性）をも解けるアルゴリズムであり，従来知られていた Arrangement Construction[7] を用いた方法よりも格段に速い．incidence graph を使うことと，我々が独自に考案した多次元 cell splitting アルゴリズムを使うことが特徴である．この結果，３次元以上の cell complex を効率よく扱うことが可能になった．

6．終わりに

Incidence graph で表現された cell complex を扱うときの課題である (1) データサイズ，(2) 空間述語，オペレーションのアルゴリズムについて Hawk's Eye での実装を述べた．

参考文献

[1] R. J. Munkers, Elements of Algebraic Topology, Addison-Wesley, Menlo Park, CA, 1984.
[2] R.H.Gűting, An Introduction to Spatial Database Systems, The VLDB Journal, vol. 3, no. 4, pp. 357--399, 1994.
[3] E. L. Gursoz, Y. Choi, and F. B. Prinz. Vertex-based representation of non-manifold boundaries. In M. J. Wozny, J. U. Turner, and K. Preiss, editors, Geometric Modeling for Product Engineering, pp. 107-130. Elsevier Science Publishers B.V., North Holland, 1990.
[4] Skiena, S. Hasse Diagrams, in Implementing Discrete Mathematics: Combinatorics and Graph Theory with Mathematica. Reading, MA: Addison-Wesley, p. 163, 169-170, and 206-208, 1990.
[5] S.Grumback, P.Rigaux, L.Segoufin, The DEDALE System for Complex Spatial Queries, Proc. ACM SIGMOD Intl. Conf. On.Management of Data, pp. 213-224, 1998.
[6] Birsson, E., Representating geometric structures in *d* dimensions: topology and Order, Discrete Comput. Geom. vol.9, no.4, pp.387-426, 1993.
[7] H. Edelsbrunner and J.O'Rourke and R. Seidel, Constructing arrangements of lines and hyperplanes with applications, SIAM J. Comput. vol.15, pp.341-363. 1986.