

24 分格子グラフによる直方体分割の描画

¹⁾ 岸良智, ²⁾ 杉田公生, ³⁾ 土田賢省, ¹⁾ 野牧賢志, ⁴⁾ 本橋友江, ¹⁾ 夜久竹夫

1) 日本大学, 2) 東海大学, 3) 東洋大学, 4) 関東学院大学

概要 - 我々は罫線保存操作のための直方体分割の表現として 24 分格子グラフを提案する。そして、24 分格子グラフに対応する H9CODE と呼ばれるデータ構造を紹介する。さらに、24 分格子グラフのセル合併の方法を提案し、描画に適用する。

Rendering 24-ary Grids for Rectangular Solid Dissections

Satoshi. Kishira¹, Tomoe. Motohashi², Kenshi. Nomaki¹,

Kimio. Sugita³, Kensei. Tsuchida⁴ and Takeo. Yaku¹

¹ Department of Computer Science and System Analysis, Nihon University

² Department of Network and Multi-Media Engineering, Kanto-gakuin University

³ Department of Mathematics, Tokai University

⁴ Department of Information and Computer Sciences, Toyo University

Abstract - We propose 24-ary grids representation of rectangular solid dissections for ruled line preserving operations. And we show data structure called H9CODE corresponding to the 24-ary grids. Furthermore, we show a volume cell unification method in the 24-ary grids. And 24-ary grids is applied to the sphere, and rendering.

1 はじめに

直方体分割はソリッドグラフィックスでよく用いられている。また、矩形分割のグラフ表現モデルとして *octrees* [1] がよく知られている。*octree* は幾何学モデリングと空間計画において使われている。また、*octree* のフォーマットは、二次元のイメージの表現のための *quadtree* 構造の拡張である。*multi-level boundary search algorithm* は、*octree* 表現で表面情報を合併するために開発されている。このアルゴリズムはグラフィックディスプレイとオブジェクト表示する課題に役立っている。我々はセルの合併のような矩形分割の罫線を保存する変形演算を考慮している。罫線を保存する変形演算を指向する矩形分割の表現として *octgrids* [2, 3, 4, 5] と呼ばれる 8 分格子グラフ、多層矩形分割の表現方法として *hexadeci-grids* [6] と呼ばれる 16 分格子グラフを提案してきた。

本論では、直方体分割に対応した *tetraicosa-grids* [6, 7, 8] と呼ばれる 24 分格子グラフを導入する。そして、計算時間が $O(1)$ であるセルの合併方法を示し、24 分格子グラフに対応した H9CODE と呼ばれるデータ構造を提案する。

セクション 2 で、準備として *octgrids* による矩形分割を述べる。セクション 3 では、24 分格子グラフの定義について説明し、セクション 4 で、24 分格子グラフに対応した H9CODE と呼ばれるデータ構造を提案する。そして、セクション 5 で 24 分格子グラフのセル合併の方法を紹介し、セクション 6 で H9CODE を用いた描画概念を示す。

2 準備

2.1 Octgrids による矩形分割

このセクションでは、不均一な矩形分割を扱い、*octgrids* [2, 3, 4, 5] の定義について述べる。

定義 2.1.1 [c.f.10]

$D = (T, P, g)$ を矩形分割とする。 T は n 行 m 列の表、 P は表 T 上の分割、 g は表 T の格子を表す。*octgrid* は D を多重辺無向グラフとする $G = (V_D, L, E_D, A_D, \alpha_D)$ で表す。ただし、 V_D は分割 P とみなす (分割 P のセル c に対応するノードは v_c により示す)。 L は辺のラベルの集合 ($L = \{emw, esw, eww, eww\}$)、 $E_D \subseteq V_D \times L \times V_D$ は無向ラベル付き辺の集合とする。ここで E_D は次に挙げるルール 1 から 4 で定義する。 $A_D = R^d$ と $\alpha_D : V_D \rightarrow R^d$ は $\alpha_D = (nw(c), sw(c), ew(c), ww(c))$ によって定義する。

ルール 1

もし $mw(c) = mw(d)$ 、即ち、セル c と d の北壁の位置が同じで、この 2 つのセルが最も近い位置にあるとき、辺 $[vc, enw, vd]$ は E_D に含まれる。この場合、辺 $[vc, enw, vd]$ は北壁辺と呼ばれる。

ルール 2-4

その他のラベル付き辺も同様に定義される。

図 1 は矩形分割とそれに対応する octgrid を表す。辺の次数が最大 8 であることに注意する。

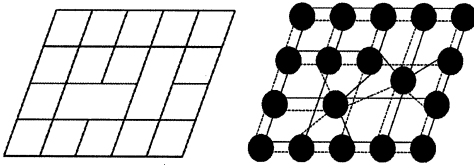


図 1 矩形分割 (左) と対応する octgrid (右)

3 24 分格子グラフ [8, 9, c.f.10]

我々は直方体分割による 24 分格子グラフを導入する。 $D = \{S_1, S_2, \dots, S_N\}$ とするとき、 D に対する 24 分格子 tetraicosa-grid は以下で定義されるラベル付き多重辺グラフ $G_D = (V_D, L, E_D, A)$ である：

(1) V_D は “直方体 $s \in D$ のとき $v_s \in V_D$ ” で定める。

(2) $L = \{EquivalentUpwardNorthEastCornerPole, \dots, EquivalentForwardCeilingNorthBeam, \dots, EquivalentBackwardFloorWestBeam\}$

(3) $E_D \subseteq V_D \times L \times V_D$ は以下で定められるラベル付き無向辺の集合：

s と t が共通の下方南側梁を持つ最も近い直方体ならば $[s, EquivalentForwardFloorSouthBeam, t]$ は E_D に属する。

(4) A は色などのボクセルの属性を表す集合である。

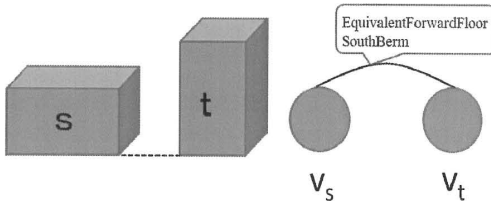


図 2 ラベル付き辺の例

図 3 は 24 分格子グラフの周辺のリンクを表す。

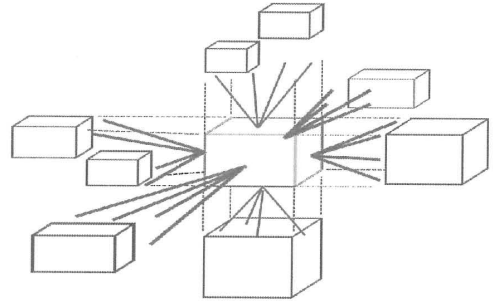


図 3 24 分格子グラフの周辺リンク

図 4 は直方体分割とそれに対応する tetraicosa-grid を表す。

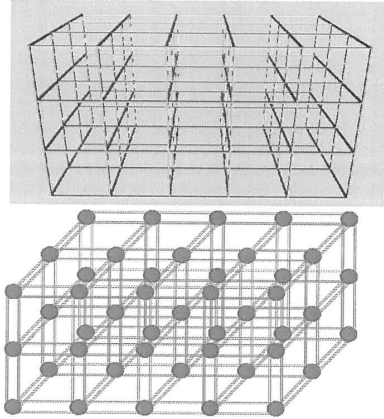


図 4 直方体分割 (上) と対応する tetraicosa-grid (下)

D を、幅 k 、奥行き l 、高さ m の直方体分割、 G_D を D に対する 24 分格子グラフ、 i を D の内部セルの個数とする。その時、
 $2 |E_D| = 12 \times 8 + 16 \times 4(k-2) + 16 \times 4(l-2) + 16 \times 4(m-2) + 20 \times 2(k-2)(l-2) + 20 \times 2(l-2)(m-2) + 20 \times 2(m-2)(l-2) + 24 i$ [7, 8]

4 データフォーマット：H9CODE

[c.f.10, 11]

我々は tetraicosa-grids に対応するデータフォーマット **H9CODE** を導入する。H9CODE は octgrids に対応するデータフォーマット H3CODE [9] に基づき拡張する。

以下は H9CODE のリスト部分であるフィールドを示す。1 ～ 32 は H3CODE と共通とする。

<共通フィールド>

1. node id
2. cell type
3. new_right
4. new_lift
5. swe_right
6. swe_lift
7. ewe_upper
8. wwe_upper
9. ewe_lower
10. wwe_lower
11. north_wall
12. south_wall
13. east_wall
14. west_wall
15. content_id
16. content_align
17. E_point
- 18-32. 予備

<拡張部分>

33. EquivalentUpwardNorthEastCornerPole
: 上方への北東角ポールの等値リンク
 34. EquivalentDownwardNorthEastCornerPole
 33. EquivalentUpwardNorthWestCornerPole
 34. EquivalentDownwardNorthWestCornerole
 35. EquivalentUpwardSouthEastCornerPole (図 5.左)
 36. EquivalentDownwardSouthEastCornerPole
 37. EquivalentUpwardSouth WestCornerPole
 38. EquivalentDownwardSouth WestCornerPole
 41. EquivalentForwardCeilingNorthBeam
: 天井側北梁の東への等値リンク
 42. EquivalentBackwardCeilingNorthBeam
 43. EquivalentForwardCeilingSouthBeam
 44. EquivalentBackwardCeilingSouthBeam
 45. EquivalentForwardFloorNorthBeam (図 5.右)
 46. EquivalentBackwardFloorNorthBeam
 47. EquivalentForwardFloorSouthBeam
 48. EquivalentBackwardFloorSouthBeam
- 以下略

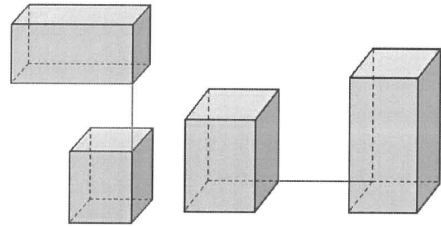


図 5 No.35(左) と No.45(右) のリンク例

5 セルの合併 [c.f.10]

我々は H9CODE を使い、適切に隣接するセル同士の合併をする。方法は図 6 のように実行される。

UNIFYVOLUME CELL (図 6)

INPUT

- G_D : 直方体分割 D の 24 分格子グラフ
 v_c : G_D のセル
 v_d : v_c と水平方向に隣接する G_D のセル

OUTPUT

- G_E : 直方体分割 E の 24 分格子グラフ

Method

1. v_d の x 軸方向のリンクの変更
2. v_d の y 軸方向のリンクの変更
3. v_d の z 軸方向のリンクの変更
4. v_d を消去

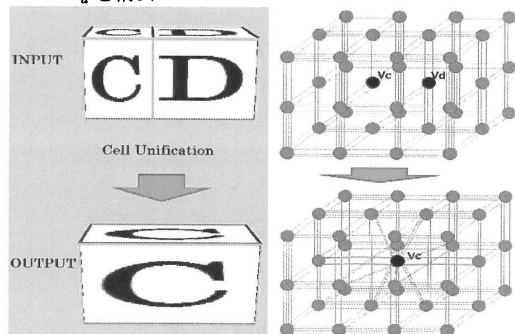


図 6 セル合併の入出力

c と d の周辺リンクの数が $96 = 48 \times 2$ に押さえられるので、アルゴリズムの計算量が $O(1)$ になることに注意する。

6 H9CODE による描画 [c.f.10]

H9CODE を用いた描画プログラムと出力のイメージを紹介する。ここではセルを複数個重ね、球などの物体を作り出す。

Program

- 座標から H9CODE を作成するプログラム
 - 名称: cv2h9
 - 入力: 座標
 - 出力: H9CODE
 - プログラミング言語: C
- H9CODE から VRML へ変換するプログラム
 - 名称: h92vrml
 - 入力: H9CODE
 - 出力: VRML
 - プログラミング言語: C

図 7 は、H9CODE を用いて生成される出力イメージを表す。

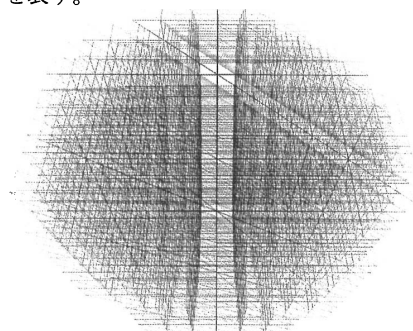


図 7 H9CODE を用いて生成される球の出力イメージ

7 おわりに

我々はボリュームグラフィックスのための新しいグラフ表現として“*tetraicosa-grid*”を紹介した。そして、*tetraicosa-grid* のデータフォーマットである H9CODE と H9CODE を用いたセルの合併方法を紹介した。今後は 24 分格子グラフのためのレンダリングシステムの実装とセルの合併の実装を構築していく。

協力いただいた有田友和氏に感謝します。

参考文献

[1] Chris L.Jackins, and Steven L.Tanimoto, Oct-Trees and Their Use in Representing Three-Dimensional Object, In Proceedings: Computer Graphics and Image, Vol.14, No.3, pp 249-270, 1980.

[2] T. Motohashi, K. Tsuchida and T. Yaku, Table Processing Based on Attribute Graphs, In Proceedings: IASTED SEA5, pp 317-322, 2002.

[3] T. Motohashi, K. Tsuchida and T. Yaku, Attribute Graphs for Table and Their Algorithms, In Proceedings: Foundation of Software Engineering, pp 183-186, 2002.

[4] 有田友和, 土田賢省, 本橋友江, 夜久竹夫, An Octet Degree Graph Representation for the Rectangular Dissections, 日本数学会応用数学分科会, 応用数学合同集会報告集, pp. 131-136, 2004.

[5] G. Akagi, T. Motohashi, K. Nomaki and T. Yaku, Octal Graph Representation for Multi-Resolution 3D Landform Maps and Its Application, 応用数学合同集会報告集, pp 27-32, 2005.

[6] A. Kureha, S. Kishira, T. Motohashi, K. Tsuchida and T. Yaku, Hexadecimal Grid Graph Representation of Multilayer Rectangular Dissections and Its Applications, 10th Society for Industrial and Applied Mathematics conf. Geometric Design & Comput, Abstract, Texas, USA, p 32, 2007.

[7] S. Kishira, A. Kureha, T. Motohashi, K. Tsuchida and T. Yaku, 24-ary Grid Graph Representation for the Rectangular Solid Dissection, 電子情報通信学会 2008 総合大会論文集, pp 171, 2008.

[8] S. Kishira, K. Tsuchida, T. Motohashi and T. Yaku, Tetra-icosa Grid Graph Representation for the Rectangular Solid Dissection, 日本応用数理学会 2008 年度 年会, pp 65-66, 2008.

[9] T. Arita, T. Yaku, H3-Code 2.3 Reference Manual, <http://www.waap.gr.jp/waap-rr/waap-rr-06-001/index.html>, 2006.

[10] T. Arita, S. Kishira, T. Motohashi, K. Nomaki, K. Sugita, K. Tsuchida and T. Yaku, Implementation of 24-ary Grid Representation for Rectangular Solid Dissections, International Conference on Computer Graphics Theory and Application (GRAPP), 2009, 予定

[11] S. Kishira, T. Yaku, H9CODE 1.0 Reference Manual, <http://www.waap.gr.jp/waap-rr/waap-rr-08-003/index.html>, 2008