

8k次格子グラフモデルによる物体の解像度最大化

夜久竹夫[†]
Takeo Yaku安齋公士^{††}
Koushi Anzai
横山隆介^{†††}
Ryusuke Yokoyama岡田直之^{†††}
Naoyuki Okada

要旨

ラスタモデル(解像度最大)を解像度低減化した矩形分割モデルの”物体”部分を、ラスタ図形に復元(“解像度最大化”)する問題を扱う。はじめに2D図形の解像度最大化(復元)を8次(疑似)格子グラフにより定式化する。次に、8次(疑似)格子グラフによる解像度最大化(復元)アルゴリズムを示す。さらに、解像度最大化(復元)アルゴリズムを用いた重なり(衝突)検知処理系のイメージを示す。

1. はじめに

移動物体の表示や物体の移動・変形・重なり(衝突)検知を同一処理系内で高速で計算することは移動物体の処理にとって不可欠である。一方物体を矩形の集まりで表すことは処理の自由度が高いため、矩形分割表現は広く用いられている。

しかしながら矩形分割表現では、表示のためには解像度が適正化された低い解像度の矩形分割が適して、一方変形や衝突検知のためには空白部分以外の物体部分が均一高解像度の図形モデルが適している。そのため矩形分割表現モデルでは、解像度低減した画像から、図形部分の解像度を最大化(復元)した画像を得る手法が必要となる。

そこで本論文では、矩形分割表現における図形部分の解像度を最大化(復元)する手法を扱う。

矩形分割のグラフモデルとして古くから矩形双対モデルが知られている。その後、探索に適したグラフモデルとして4分木モデルが提案されて[1]、さらに4分木由来の8分木[2]、16分木[3](本論文では総称して”4k分木”という)が提案されている。矩形双対グラフや4k分木モデル上で重なり検知アルゴリズムや結合を通常の方法で構成すると計算時間は最悪 $O(n^2)$ になることに注意する。矩形分割に関しては他に、罫線指向型処理に適した8次格子モデルが提案されて[4]、その後一般的な8k次格子グラフが提案された[6]。8次格子グラフでは結合などいくつかの処理が $O(n)$ 時間で解ける[5, 6]。また重なり(衝突)検知では1x1セルの集まり(解像度最大)からなる図形が速いアルゴリズムを導くことが示されている[7]。

そのため、8k次格子グラフでは速い解像度最大化(復元)アルゴリズムが処理系全体の処理を速くする可能性がある。

そこで、本論文は次の事を目的とする：(1) 8k次格子における解像度最大化(復元)の定式化、(2) 2D物体の解像度最大化手法、(3) 解像度最大化(復元)の利用。結果は次の通りである：(1) 8k次格子による解像度最大化(復元)の定義、(2) 2次元物体の8次格子による $O(n)$ 時間の解像度最大化(復元)アルゴリズム、(3) 解像度最大化(復元)を用いた処理系のイメージ。

結果は重なり検知や移動物体の衝突検知に応用される。2節で解像度最大化の定式化、3節で解像度最大化アルゴリズムの概略、4節で応用した処理系のイメージを示す。

2. 解像度最大化(復元)とその利用

2.1 8次格子:矩形分割のためのデータ構造[4]

定義 2.1 グラフ G が矩形分割 D に対する8次格子 $\leftarrow \text{def} \rightarrow$ 2頂点 c と c' は c と c' が同じ罫線を共有する時に”リンク”される(図1参照)、ただし便利のために最も外側の矩形のサイズは1x1とする。3D, 4D, 複数図形の場合も同様に定義。

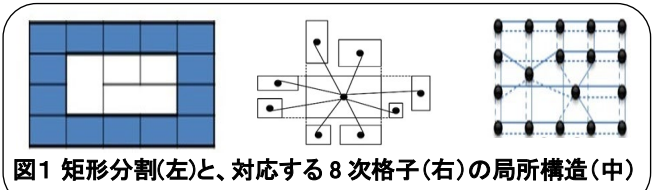


図1 矩形分割(左)と、対応する8次格子(右)の局所構造(中)

2つの物体は2つの層で表される[6]。各頂点には位置(“東西南北辺”の”罫線番号”4個)が付随する。

2.2 解像度最大化(復元)

以下の8次格子を考える：全ての頂点は黒(占有)か白(空白)で色付け。各物体は黒の頂点の集まりで表現される。8次格子の解像度最大化(復元)を8次格子の中の専有部分の頂点のサイズを全て1x1にする(図2参照)ものと定義する。定義は8k次格子により複数図形、3D, 4Dに一般化される。

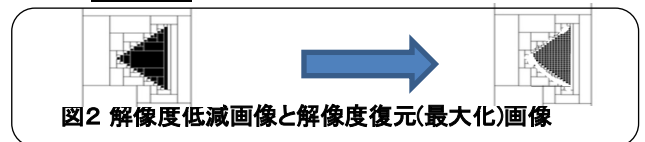


図2 解像度低減画像と解像度復元(最大化)画像

3. 8k次格子モデルによる解像度最大化(復元)

3.1 アルゴリズム

次のアルゴリズムは、高さ2以上のセルに対応する頂点 v (図5(1)、図4左)を高さ1に対応する v と高さ1以上に対応する v' (図5(2)、図4右)に水平分割する。ここで、 s, s' はそれぞれ v の西側で v に最も近い南壁共有頂点と v' の西側で v' に最も近い北壁共有頂点とする。 s, s', v, v' を後の解像度最大化(復元)アルゴリズムで利用する(*)。

Algorithm *CellHorizontalSplit*(v, G, s, s', v', H)

DATA

INPUT $G=(V, E)$: $n \times m$ の矩形分割 D に対応する8次格子;
 v, v' : 上の頂点 (*)

OUTPUT $H=(V', E')$: G で物体領域の解像度が最大化された矩形分割 F に対応する8次格子, in which the node v is splitted to v and v' .

METHOD (outline) 図4左を考える。

/* はじめに v を v, v' に分割 */

① $v-qE$ 辺を $v-v'$ 東壁辺と $v'-qE$ 東壁辺で付け替える;

② $v-qW$ 辺を $v-v'$ 西壁辺と $v'-qW$ 西壁辺で付け替える;

/* 図3中が得られる */

/* 次に u の南壁辺を処理 */

[†] 日本大学 Nihon University

^{††} 関東学園大学 Kanto Gakuen University

^{†††} 応用オートマトン研究会 Workshop on Automata and Their Applications

- ③ s の南壁辺で東隣を t とする;
- ④ s - t 南壁辺を s - v 壁辺、 v - t 南壁辺に付け替える;
/*次に v 'の北壁辺を処理*/
- ⑤ s' の北壁辺で東隣を t' とする;
- ⑥ s' - t' 辺を s' - v' 北壁辺、 v' - t' 北壁辺に付け替える;
/*図3右と図4の2番目が得られる*/プログラム終了

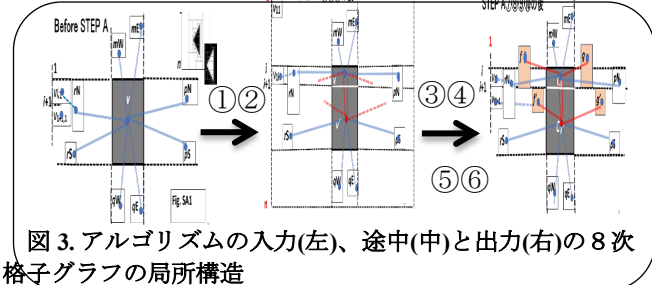


図3. アルゴリズムの入力(左)、途中(中)と出力(右)の8次格子グラフの局所構造

同様に垂直分割のための Algorithm CellVerticalSplit(v, G, x, x', v', H)を構成可能である。頂点次数が8以下のため、

命題1. 上のセル分割は定数時間で実行可能。
次に解像度を最大化するアルゴリズムを考える。ここでは矩形分割の中の一つの大きなセル(図5(1)上左)は、1 x 1のセルの集まり(図5(7)下右)に分割される。

Algorithm 2DResolutionMaximizationWith8grid(G, H)

DATA

INPUT $G=(V, E) : n \times m$ の矩形分割に対応する8次格子

OUTPUT $H=(V', E') : G$ で物体領域の解像度が最大化された矩形分割に対応する8次格子

補助変数

NearestWNWallNode, NearestWSWallNode: サイズ n の配列
NearestNEWallNode, NearestNWWallNode: サイズ m の配列
 WN, WS, NE, NW と略す

METHOD

Initialize $WN(i), WS(i) \leftarrow v_{i,1} (i=1, \dots, n)$;
 Initialize $NE(j), NW(j) \leftarrow v_{1,j} (j=1, \dots, m)$;
do $i=1$ **to** n
 {はじめに $v_0 \leftarrow V_{i,1}$ とし、 V_0 から北壁辺を東に辿り、最初の高さ2以上の頂点を v とする;
 v の西側で v に最も近い北壁(南壁)共有頂点を、 $WN(i)(WS(i))$ から探索して新たな $WN(i)(WS(i))$ と置く;
 Algorithm CellHorizontalSplit($v, G, WN(i), WS(i), v', H$)を適用し、出来た H を新たな G とする};
 同様に Algorithm CellVerticalSplit($v, G, NE(j), NW(j), v', H$)を繰り返し適用して垂直分割を行う;
 全ての行と列について実行したら終了

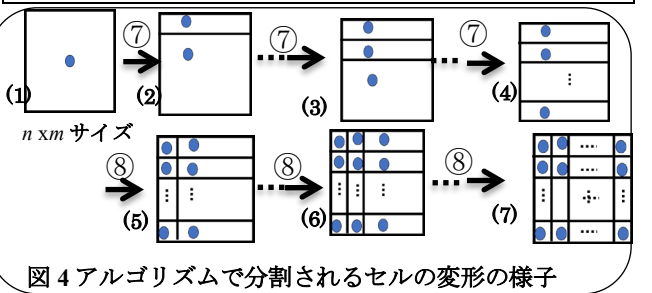


図4 アルゴリズムで分割されるセルの変形の様子

3.2 アルゴリズムの性質
NearestWNWallNode, NearestWSWallNode, NearestNEWallNode, NearestNWWallNode の探索は全体でそ

れぞれ1パス。頂点に位置を入れるのに4パスだから、全体で8パス。従って命題1から:

定理1. 上の 2DResolutionMaximizationWith8grid (解像度最大化(復元))は $O(n)$ 時間で実行される。

4. 解像度復元の応用例

図5のような処理系を考える。データ構造に8次格子を用いると解像度低減による矩形数が最良で4分木の半分程度になる。表示時間は矩形数に依存するから4分木モデルより8次格子モデルの方が表示には適している。しかし、図形の特徴抽出や変形処理は均一解像度(解像度最大)の方が短時間で実行可能である。従って表示/変形を繰り返すカーナビ等の処理系では解像度低減画像と解像度復元画像を繰り返し使用するのが便利であり、そのために効果的な解像度復元処理が必要となる。

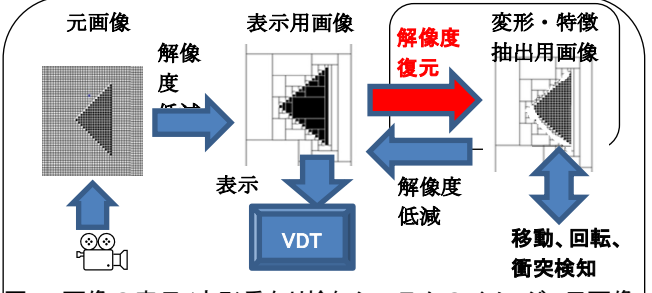


図5. 画像の表示/変形重なり検知システムのイメージ。元画像(左)、解像度低減画像(中上)、図形部分解像度復元画像(右)

5. おわりに

我々は以下を導入した: 8k次格子における解像度最大化(復元)の定義 (§2), 8k次格子による $O(n)$ -時間の解像度最大化(復元)アルゴリズム (§3.1, §3.2) の概要, 解像度最大化(復元)を応用した変形・衝突検知処理のイメージ (§4)。これらにより変形・衝突検知などの処理が従来より低い時間計算量で実行される可能性がある。また、表計算のセル分割にも応用可能である。今後は高次の場合(cf. [3, 6])と実装を考える。土田賢省氏, 山崎浩一氏, 水野房江氏に感謝します

参考文献

- [1] R. A. Finkel and J. L. Bentley, Quad Trees: A Data Structure for Retrieval on Composite Keys, *Acta Inform.* 4, 1-9, 1974.
- [2] C. L. Jackins and S. Tanimoto, Oct-trees and their use in representing three-dimensional objects *Computer Graphics and Image Processing* 14(3), 249-270, 1980.
- [3] N. Inamoto, Hexadecimal-Tree: A Time-Continuous 4D Interference Check Method. In: Falcidieno B., Kunii T.L. (eds) *Modeling in Computer Graphics. IFIP Series on Computer Graphics*. Springer, 1993.
- [4] T. Yaku, Representation of Heterogeneous Tessellation Structures by Graphs, *Mem. WAAP Meetings* 108, 6p, 2001. URL <http://www.waap.gr.jp/waap-rr/waap-rr-01-013.pdf>
- [5] G. Akagi, K. Anada, S. Koka, Y. Nakayama, K. Nomaki, T. Yaku: A resolution reduction method for multi-resolution terrain maps. *SIGGRAPH Posters* 2012: 86, 2012.
- [6] T. Yaku, K. Anada, K. Anzai, S. Koka, Y. Miyadera and K. Tsuchida, 8k-ary Grid Graph Models of Tabular Forms, *Springer LNCS* 8373, 465-477, 2014.
- [7] 夜久竹夫, 安齋公士他, 8k次格子グラフモデルによる物体の重なり検知, FIT2024, CA-005, 2024