セル構造オートマトンによる図形生成について

5S－5

荒田秀樹　　高井昌彰　　佐藤義治
北海道大学 工学部

1　はじめに

セル構造オートマトンは、現在のセルの状態と予め定義された近傍の状態により次の時刻のセルの状態を決定する局所的計算モデルで、比較的簡単な近傍形及び状態遷移規則の下で全体として複雑な挙動を示すことが知られている [1]。また、本質的に並列アルゴリズムであり、マルチプロセッサや分散システムとの関連性も高い。

本稿では、ピクセル空間を 2 次元のセル空間と考え、セル構造オートマトンを用いて幾何図形の生成を行う。セル空間上で局所的な情報を伝達するための手段としてある種の波を発生させ、その情報を活用してセル空間の大域的制御を試みる。

2　図形生成モデル

一般にピクセル空間上に線分などの幾何図形を描く場合、端点の位置、直線の傾きなどの情報を一元的に管理する上位機構が存在し、ユーザから明示的に与えられた情報をもとに図形生成を実現している。この際、ピクセル空間は受動的なメモリであり、図形生成はそれに対する逐次的処理である。これに対し、処理機能を有する能動的な実体としてピクセル空間をとらえ、その上で幾何図形を並列的かつ自律的に発生させるアプローチは大変興味深い。

そこで我々は、ピクセル空間を 2 次元のセル構造オートマトンと見なし、図形の基本的要素である端点、線分を自律的に発生させ、その集合として図形を生成する手法を提案する。ただし、すべてをセル構造オートマトンで実現することは難しいので、ユーザから与えられる幾何学的な制約条件を考慮して図形生成を行う。すなわち、端点発生、線分発生などを実現する基本的な状態遷移関数を適切に組み合わせることで制約条件を満たす目的の図形を生成する。

例えば三角形 ABC を描く場合、ユーザは" 端点数 3 の多角形" という制約をピクセル空間に与える。セル構造オートマトンは、この制約を解くために 3 つの端点をピクセル空間上の任意の場所に発生し、続いてそれらを結ぶ線分を生成する。

2.1　セルの状態

図形生成のために図 1 のような、1 枚の可視プレーンと、発生させるべき端点の数と等しい数の不可視プレーンを考える。端点の数は、制約条件として最初に与えるものとする。可視プレーンにおいて、セルは 2 つの状態をとりピクセル値を表す。また、セル自身の座標と ID 、他の端点候補の座標と ID 、端点を表すフラグ、線分生成の波の発生を表すフラグを属性として持つ。不可視プレーンでは、セルは 2 つの状態をとり端点又は線分であるかどうかを表す。また、他の端点の座標、自分と端点との座標の差分、自分と端点との距離、2 端点間の傾きの積算値を属性として持つ。

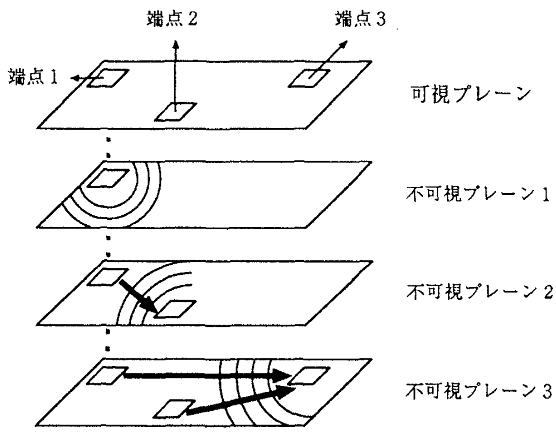

図 1: 線分生成の概念図

2.2　図形生成の手法

可視プレーンでは端点の発生及び線分の描画を行い、不可視プレーンでは、それぞれのプレーンに該当する端点間を結ぶ処理を行う。以下に端点の発生、端点間を結ぶ処理、線分の描画について説明する。

Geometrical Image Generation By Cellular Automata
Hideki Arata,Yoshiaki Takai,Yoshiharu Sato
Hokkaido University,Sapporo 060 JAPAN.

2.2.1 端点発生

端点の位置は、ユーザから明示的に与えられない限りピクセル空間上で任意である。したがって、ピクセル空間から任意に 1 つのセルを選び出す機能が必要となる。これは、分散システムにおけるリーダ選出問題と同様である。

まずすべてのセルを端点候補とする。すべてのセルは自分の強さを示す ID を持っているので、その ID と自分の座標値を波としてセル空間に伝搬させる。他の端点候補の波と衝突した場合は ID を比較し、強い方のセルの波を更にセル空間に伝搬させる。この手順を繰り返すことで 1 つの端点を発生させることができる。

複数の端点を発生させる場合は、すでに発生させた端点を除いて同様なルールを適用すれば問題なく発生させることができる。

2.2.2 線分発生

端点が発生次第その端点について線分発生の処理を行う。端点の発生と同時にその端点の座標値をピクセル空間全体に伝搬させる。他の端点がその座標値を受けとると、座標値の差を計算し、それを 2 点間の傾きとしてとらえる。この傾きを、線分の発生の過程における増分値として考える。本手法は、DDA(digital differential analyzer) [2] を参考にしている。DDA では直接ピクセルの座標を計算するのに対し、本手法では座標は計算せず、傾きの積算値を伝搬させながら自分が線分になるべきか否かを局所的に決定している。直線発生の方向及びそれぞれのプレーンでの該当する端点は図 1 のようになる。

線分発生の例として、傾きが 2/7 の場合を図 2 に示す。この場合、まず端点 A から右方向に直線を発

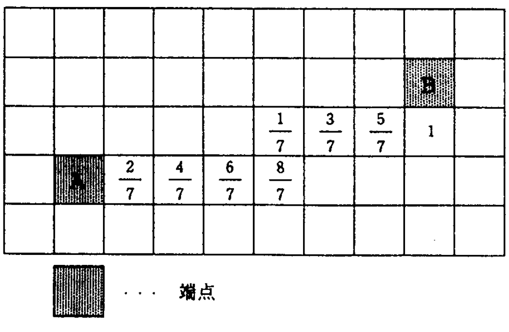

図 2: 端点 AB 間の直線発生

生し、傾きの積算値が 1 を超えるまで直線を延長していく。積算値が 1 を超えると、上方向に向きを変え、積算値から 1 を引く。以後同様なルールを繰り返せば端点 B に達する直線を発生することができる。

線分の描画については、各不可視プレーンで発生した線分をすべて可視プレーンに投影すれば、目的の図形を得ることができる。

3 実行例

本モデルで生成した簡単な幾何図形の例を示す。図 3 は、256×256 のセル空間に 3 つの端点を発生させ、それらを結んで三角形を生成したものである。原理的には、1 ピクセル = 1 処理ユニット構成の超並列計算機への実現が可能である。

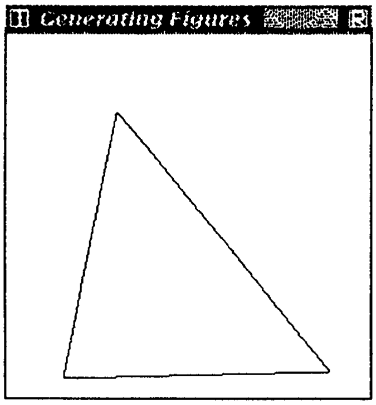

図 3: 三角形（端点数 3 ）の生成

4 まとめ

ピクセル空間を 2 次元のセル空間と考え、セル構造オートマトンを用いて図形生成を行うためのモデルを提案し、そのモデルを用いてセル空間の大域的制御の可能性が確かめられた。

現在本モデルを用いて、端点発生、直線発生に加え、一定長さの線分や円の生成が可能である。より複雑な幾何学的制約条件を用いた図形生成が今後の課題である。

参考文献

[1] T. Toffoli and N. Margolus: *Cellular Automata Machines*, MIT Press (1987).

[2] D. Hearn and M. P. Baker: *Computer Graphics*, Prentice Hall (1986).