2008－MUS－76 (3)
2008／8／7

ライフゲームのセルパターンを対応する音に変換する ライブ音楽生成システムの試み

小川 圭祐† 久原 泰雄†

† 東京工芸大学芸術学部メディアアート表現学科

本システムではライフゲームにおいて複雑に変化するパターンを実演者が操作することによって作成される音楽を目と耳で楽しむことができる. 実演者はライフゲームのマトリクス上に生み出される 2 次元セルパターンに対して発音の要素を割り当てる. システムは刻々と変化するマトリクス上に定義されたセルパターンを検索し, 合致したらそのパターンに対応する音を生成する. 合致したパターンの座標情報はルート音, オクターブ, ベロシティなど発音に関係する属性として用いる. 実演者はマトリクス上の生命をマウスで入力可能なので, インタラクティブに意図するパターンを入力し, 楽曲生成に介入することができる. ライフゲームの世代更新は実演者が定義したタイミングで行われ, 楽曲のグルーブ感を形成する. 複数のマトリックスを同時に稼動させ, マトリックス毎に, パターン割当て, 更新タイミング, 音色を設定し, アンサンブルを行うことによってライフゲーム・オーケストラを構成する. ライフゲーム・オーケストラは, 実演者の意図とセルオートマトンの創発的なパターン変化の融合を目指したシステムである.

The Trial of Live Music Composition System Translating the Cellular Patterns of Conway's Game of Life into the Synchronized Sounds

Keisuke OGAWA† Yasuo KUHARA†

† Department of Media Art, Faculty of Arts, Tokyo Polytechnic University

Our system generates music by dynamically assigning cellular patterns of Conway's game of life to the synchronized sounds. A performer can compose by controlling the complexly varying patterns and the sounds with visual and auditory fun. A performer assigns the elements of tone generation to two dimensional cell patterns in the matrix of the game of life. Our system searches the defined cell patterns in the dynamically varying matrix. If it can match the pattern, the tones corresponding with them are generated. The data of coordinates of the matched patterns are used for the tone elements like as root, octave, velocity, etc. A performer can make a life in the matrix by a mouse drag, and influence to the generating music interactively. The generation is updated by the timing defined by a performer to configure the groove of music. By running multiple matrices with different pattern mapping, update timing, and instruments, their ensemble has become the life game orchestra. The life game orchestra is the fusion system of the design of a performer and the emergence of cellular automata with various pattern transitions.

1. はじめに

本システムはセルオートマトンの一種である Conway のライフゲームを用いて音楽を生成する. セルオートマトンを用いた音楽生成は, セルの状態をピッチ, 音長, 音色などに適用した例[1][2]に始まり, WolframTones[3], CAMUS[4], glitchDS[5]など様々なシステムが考案されている. たとえば glitchDS では, ライフゲームはシンセサイザーの音を制御する楽譜として機能し, 電子音を歪ませ加工するグリッチサウンドであり, 音程が明確でない音列を利用し, 独特のリズムサウンドを生成する.

ライフゲームのような二次元マトリクスにおける創発的な状態遷移は, 自律的に音を構成する作曲において有効な情報源である. しかし複雑な遷移から生み出される情報の処理は容易ではなく, 仮にすべてのセルの座標や状態の情報を音符に置換するならば, 発音数が拡散し, これらを制御し音楽を構成するのは困難である.

本システムでは, ライフゲームは個々のセルの状態遷移で構成されながらも, マクロな視点では2次元のセルパターンの集合として認識されることに注目した. つまり周期図形のブリンカーやグライダーのように特定のセルパターンが観察者の目を引き, その動きがライフゲームの面白さとなっているので, セルパターンを発音の基本要素とした. 2 次元マトリクス内に発生する小さい単位のセルパターンを発音のトリガーとし, 音程やテンポなどの音楽の構成は実演者が制御することによって, ライフゲームの持つ予測不可能な創発と実演者の自立的な音楽構成を融合した音楽表現を試みた. またライフゲームのマトリクスを複数同時に進行させ, マトリクスごとに音色, テンポなどを割り当ててアンサンブルを行わせた.

2. 方法

ライフゲームの計算処理を行う Max/MSP+Jitter の jit.conway オブジェクトを使用した(図 1 参照). 使用した 2次元マトリクスは 16×16 の 256 セルで構成される. このマトリクス上で進行するセルの状態遷移を離散時間ごとにパターンを解析し, パターンが実演者の意図するパターンとマッチングしたとき, 発音を促す. また現時点では, 生成される楽曲は MIDI によって発音される.

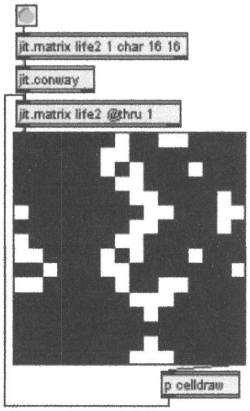

図 1 jit.conway によるライフゲームの 16×16 マトリクス 白が生で黒が死を表す.

2. 1 セルパターンと音のマッピング

ライフゲームには頻繁に出現する小さなセルパターンをいくつも確認することができる. これらのパターンは別のパターンと干渉しカオス的に拡大したり, 死滅に至ったり, 頻繁に成長と衰退を繰り返す. その際, 観察者はセル1つ1つの状態を認識しているのではなく, セルを 2 次元の集合として認識している. セルの集合をどこまで一塊のパターンとして認識するかは人によって異なるが, セルパターンの発生と消滅に視覚的な面白さがある. アルゴリズム作曲において乱数的な要素を制御することは重要な関心事である. ライフゲームの状態遷移は局所的ルールに基づいているため決定論的であるが, マクロな視点からセルパターンの成長の推移と見ると予測不可能であり, また創発的である. これらの要素を音の生成の素材として用いるとセルパターンの発生に応じて曲が変化し, 創発的な視覚情報を音楽情報に変換していることになる. このような発想の元, 図 2 にはセルパターンとスケールを対応付けた例を示す.

2. 2 セルパターンのマッピング・ポリシー

特定のセルパターンをどのようなスケールにマッピングするかは本システムにおいて生成される楽曲を左右する重要な要素であり, また実演者の楽曲構成の手腕が試される所でもある. 複数の音の関係により協和音や不協和音が生まれ, さらに音階, 旋法, 和声が生まれる. さらに不協和音を巧みに使うことによりテンションを伴った音楽を構成することは一般的な作曲手法である. スケールに添っているか, または外れるかといった変化は, ライフゲームの世代更新に伴うセルパターンの発生率に基づいて割り当てるスケールを対応付けることによって実演者の意図する音楽を構成することができる.

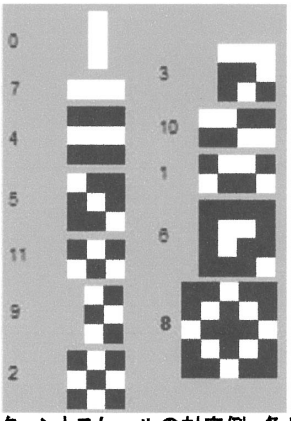

図 2 セルパターンとスケールの対応例. 各セルパターンの左にルート音からのインターバル(半音を 1 とした数値)を記した. 発生しやすい単純なパターンに協和的なインターバルを割り当てた.

単純なパターンは発生しやすく, 複雑なパターンは発生しにくいと推測できる. したがって発生しやすいパターンを発生させたい音の発音として機能させ, 発生しにくいパターンは発生させたくない音の発音に利用できる. 結果として, 生成される楽曲に調性を持たせることが可能となる. 曲の展開は, セルパターンに依存するので, 様々な音楽スタイルをセルパターンのマッピングにより設計することができる. またブリンカーやグライダーといったライフゲーム特有のパターンは連続して発生する. このような特徴を考慮してセルパターンと音のマッピングを設計することができる. あるいはこの特徴を無視してマッピングすることで, 結果的に興味深い作曲に繋がることも考えられ偶然性を利用することも期待できる.

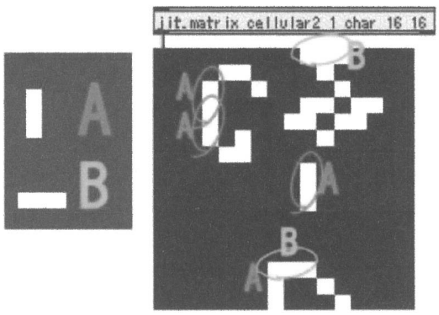

図 3 セルパターンのマッチングの例. パターン A および B が検索されている.

2．3 セルパターン・マッチング

現時刻のマトリクス内に実演者が指定したセルパターンを検索する．セルパターンが合致したら，割り当てられた音を鳴らす．また合致したセルパターンの座標を出力し，ルート音，オクターブ，ベロシティなどに利用する．図3は，パターン A と B がある時刻におけるマトリクス上で検索された例である．パターンマッチング処理は Max/MSP の js.オブジェクトを使用し，Javascript によって記述することで処理の高速化を図った．

2．4 世代更新のタイミング

ライフゲームの世代更新によるセルの状態遷移は音楽の進行に相当し，世代更新のクロック制御は，発音のタイミング制御となる．通常のライフゲームでは世代更新のタイミングは一定の時間間隔において行われることが多く，一定のテンポで発音をする．一方，本システムでは，世代更新のタイミングを実演者が制御することによって，楽曲のテンポをリアルタイムに変化させることができる．たとえば，協和音による調和の取れたスケールに割り当てたセルパターンを発見したときは，メロディを意識させるために世代更新のタイミングを遅くする．不協和音のパターンが発見された場合は長く響かせないために世代更新のタイミングを速くしたりする．こうして実演者の意図に合う楽曲を構成することができる．また，セルパターンが多量にマッチングした場合，音が過密になっているので，更新タイミングを遅くする演出を行うとライフゲームの進行と生成される音楽に視覚的，音楽的な興味深い変化を及ぼす．なお，タイミング制御には Max/MSP の matrixcontrol オブジェクトを利用した(図4参照)．

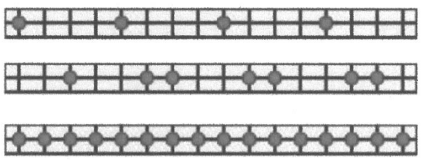

図4 matrixcontrol による世代更新タイミング制御．実演者がマトリクスごとにタイミングを設定することで，生成される楽曲のグルーブ感を構成する．

予め複数のリズムパターンを設計しセルパターンと対応付ける．マトリクス上にセルパターンがマッチングしたとき，それに該当するリズムパターンを matrixcontrol に読み込む．たとえばマトリクス内にセルパターンが過密状態の場合に長い音価のリズムパターンを適用し，過疎の場合には細かい音価のリズムパターンを適用するといった演出である．これはライフゲームのセル状態遷移に依存したグルーブ感を生成していることになる．

2．5 実演者によるセル状態の入力

本システムのコンセプトの一つに誰もが作曲演奏を楽しめると同時に，作曲や音楽の習熟度が演奏に反映されることあげられる．インターフェイスは簡潔で直感的なものが理想である．そこで実演者がリアルタイムにマウスドラッグする方法が妥当であると考えた．実演者はライフゲームのマトリクス上をマウスドラッグすることによりリアルタイムにセルパターンを描くことができる．セルパターンによってスケールが定義されているため，実演者はセルパターンの発生率を考慮しながらドローイングする．常にクロックは進行している状態なので，自動的に世代更新が行われる．リアルタイムでセルを書き換える行為により，生み出される音楽の変化を制御できるので，ライフゲームによる楽曲生成に実演者が介入して，楽曲生成に変化を加えるインタラクティブな作曲装置として機能する．

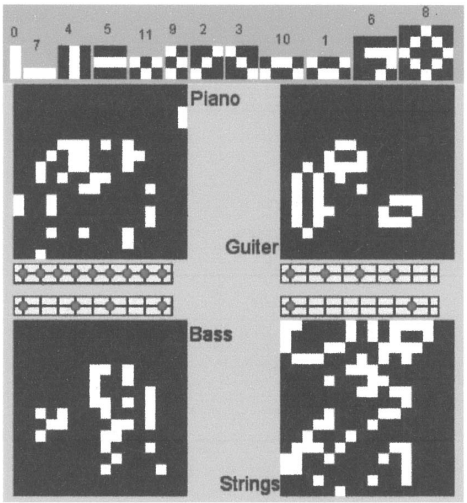

図5 ピアノ，ギター，ベース，ストリングスの4パート構成のライフゲーム・オーケストラの例．各パートに独立した世代更新タイミングが matrixcontrol によって定義されている．上部にはセルパターンに対応付けられたスケールが記されている．

2．6 複数マトリクスのアンサンブル

複数のライフゲームのマトリクスを同時に進行させることにより，アンサンブルを行う．この際，個々のマトリクスに別々の音色，別々のセルパターン・マッピング，別々の世代更新タイミングなどを適用する．それらの組み合わせを複数用意して，それぞれ違ったパラメータを試せば，変化に富んだアンサンブルを行うことが可能である．こうしてライフゲーム・オーケストラが構成される(図5参照)．

ライフゲーム・オーケストラにとって，各マトリクスの音色の選択は，セルパターンの発生率やリズムパターンの特性を考慮することが理想的である．例えば細かい単位でリズムが変動するようなマトリクスの場合，アタック音が強く，ゆっくりと減衰するピアノのような音色が好ましい．あまりリズムがゆれないマトリクスの場合，減衰の早いマリンバやパーカッションのような音色が向いている．しかし

すべては実演者の発想により自由に選択できる．また楽音にとどまらずノイズやサンプリングデータなどを使用することもできる．

それぞれのマトリクスの最端辺を別のマトリクスの最端辺に適時対応させれば，各マトリクス同士のセルの状態遷移に相互作用を与えることができる．この場合は，複数のマトリクスが1つのマトリクスとしてセルが行き来しながら，それぞれが別のタイミングで，別の音色で，別のスケールによって演奏を行うことができる．

2．7 調性の変更

前述の通り，実演者はマウスドラッグにより，セルパターンを描いて，音楽を演奏することができる．このとき，セルパターンに基づいて発音が成されるが，発生率を加味したセルパターンとスケールをマッピングした場合，実演者の意図する調和の取れたメロディが自動生成されることが期待できる．

しかし演奏を行っていれば，音楽として調性を展開させたいという欲求が出てくる．そこでマウスでリアルタイムにスケールのルート音を変更することにより，調性を制御できるように設計した．例えばマトリクスを縦に 12 分割して，左から C, C♯, D, ・・・, B という様に調性を設定する．例えば C の列をマウスクリックしたあとに，D の列をクリックするとルート音が C から D に変更され，調性が変化する．同様にマトリクスを横にも分割して，長音階や短音階を決定づける音に対応しているセルパターンを置換すれば調性が変化する．

このようにしてマトリクスの領域に応じたルート音や調性の制御はマウスによるセルパターンのドローイングと同時に行われるので，ある場所でセルを描きながら曲調を変化させたり，曲調を変化させるためにある場所にセルを書き込んだり，といった操作を行うことになる．

3．結果と考察

発生しやすいセルパターンからなされた発音は細かい分散したメロディを作り上げた．結果として，セルの動きを音で表したような視覚的な要素が直接的に音楽に変換されたような楽曲となった．セルパターンとスケールのマッピングによる発音の仕組みは調性を維持し，楽曲の調和を成立させている印象を抱かせた．

作曲に造詣の深い実演者が音楽理論に基づいてセルパターンのマッピングを設計することにより，複雑な音楽を実現することができる．一方，音楽を知らない実演者でもセルパターンというグラフィカルなインターフェイスによって，個性的な作曲が行える．セルをリアルタイムでドローイングし，そのドローされた座標によりコード進行などを変える演出は，より実演者の意図を表現する手助けとなった．

なお参考のため，実演のデモンストレーション例の動画を Web サイト[6]に紹介する．

4．課題

まずライフゲームの世代更新は複雑系であるものの，実演者によってまったく違う進行になることはありえない．初期値が同じならば，当然同じ進行になる．演奏中の実演者の介入や，事前の音階の設定やセルパターン・マッピングと発音の割り当てにより，ある程度の実演者の独創性を追求することもできるが，セルの進行がオートマトンのルールに従う限り，実演者の演出はライフゲームの世代更新によって，減衰してしまう．意外性のある音楽生成という面白さがある反面，実演者の創造的な余地はあまり見られないのが現状である．

これらの問題の解決方法のひとつとして，出現するパターンの意味を作曲者が深く理解し，パターンの出現を予想しながら制御するといった提案が浮かぶ．予想困難なライフゲームの世界でそれらをリアルタイムに予想し制御するというのは困難だが，ある程度の熟練により，予測が立てられれば，複雑な偶然性を上手く活かした作曲を展開することができるかもしれない．

5．展望

ビジュアル面の改善を検討している．ライフゲームのセルの状態遷移は視覚的に魅力的なので，音と色彩の対応などよりグラフィカルなインターフェイスの設計を行いたい．また入力インターフェイスの改善も必要である．現時点ではマウスのみで入力しているが，タッチパネルを用いれば操作性は向上する．ライブカメラやプロジェクターを利用して空間上で指揮をするように音楽を統制できるインターフェイスも考慮中である．今回はライフゲームのセルパターンに主に注目して研究を行ったが，今回の試みで使わなかったパラメータはいくつもある．これらを音楽生成の要素として結び付けることよって，誰もが創造性を発揮し表現できるシステムを構築することを目指していきたい．

参考文献

1) Beyls, P, "The Musical Universe of Cellular Automata", Proceedings of International Computer Music Conference, pp.34-41, 1989.
2) Millen, D "Cellular Automata Music", Proceedings of International Computer Music Conference, pp.314-316, 1990.
3) http://tones.wolfram.com/generate/
4) http://tamw.atari-users.net/camus.htm
5) http://www.glitchds.com/
6) http://www.media.t-kougei.ac.jp/lifegorch/