一般社団法人 電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report

フラクタル/カオスのライヴ生成に関する高速化の検討

長嶋 洋一†

†静岡文化芸術大学 〒430-8533 静岡県浜松市中区中央2-1-1

E-mail: †nagasm@suac.ac.jp

あらまし フラクタルやカオスを応用した「数理造形」デザインに関して、筆者は2022年1月に「触覚/触感インターフェースとライヴ生成フラクタル/音響によるウェルネス・エンタテインメント」という報告を行った。本稿ではその続編として、Max8環境において詳細な動的フラクタル・グラフィクス生成をスムーズ/高速に実装するためのGENという枠組みの適用について検討し、具体的なシステム/作品の試作として追求した実験について報告する。

キーワード フラクタル, Max8, Gen, ライヴ・グラフィクス

On a method to speed up live fractal/chaos generation

Yoichi NAFASGIMA†

†Shizuoka University of Art and Culture 2-1-1 Chuo, Naka-ku, Hamamatsu, Shizuoka, 430-8533 Japan

E-mail: †nagasm@suac.ac.jp

Abstract Regarding "mathematical modeling" design applying fractals and chaos, the author reported in January 2022 on "Wellness Entertainment with Tactile/Haptic Interface and Live Generated Fractals/Acoustics". As a sequel to that report, this paper discusses the application of the GEN framework for smooth and fast implementation of detailed dynamic fractal graphics generation in the Max8 environment, and reports on the experiments pursued as prototypes of concrete systems/works.

Keywords Fractal, Max8, GEN, Live Graphics

1.はじめに

筆者はこれまでもNLP研究会などの場でフラクタルやカオスを応用した「数理造形」デザインに関して報告してきた[1-12]。色々なセンサをメディアアートのヒューマンインターフェースとして活用するという筆者の研究の領域では、新しい「触覚/触感センサ」であるPAWセンサを活用すると共に、これが人間の「内受容感覚」に働きかけるバイオフィードバックによって「ウェルネス・エンタテインメント」(→福祉領域)に通じる、という可能性を検討し報告してきた[13-22]。センサ情報に対応して、サウンドとグラフィクスをリアルタイム生成するシステムの開発プラットフォームは、筆者は1991年にプロトタイプと出会い、32年間ずっと活用している「Max」(現行バージョンはMax8)である。Maxについての解説は末尾の参考文献URL[23-36]を参照されたい。本稿では、筆者が2022年1月に報告した、フラクタルやカオスを応用した「数理造形」デザインに関する「触覚/触感インターフェースとライヴ生成フラクタル/音響によるウェルネス・エンタテインメント」の続編として、Max8環境において詳細な動的フラクタル・グラフィクス生成をスムーズ/高速に実装するためのGENという枠組みの適用について検討し、具体的なシステム/作品の試作として追求した実験について報告する。

2.Maxのライヴ・グラフィクス機能の歴史

筆者が共立出版「bit」別冊「コンピュータと音楽の世界」の編著に関わり、「アルゴリズム作曲」の章[37]でMaxのプログラミングについて解説したのは1997年のことであるが、ここでplain text紹介したMax(version 2)パッチ[38]は、現在のMax(version 8)にコピペしてもそのまま同等に走る。この優れた上位互換性は卓越したビジュアル・プログラミングの構想の賜物であるが、そのシステムの起源は、フランス国立の音楽音響研究所IRCAM[39]にある。NeXTコンピュータに挿入するリアルタイム音響信号処理ボードISPW(IRCAM Signal Processing Workstation)のためのプログラミング環境MAX/FTSであり、開発した研究者: David ZicarelliとMiller Puckette の師匠であるMax Mathews氏(Stanford音楽研究所CCRMAおよびIRCAMの教授: Computer Musicの研究者として半世紀以上、世界をリードした)からその名をとったものである。

このようにMaxの起源はリアルタイム音楽情報処理にあるが、マルチメディア・パフォーマンスでもある音楽にとってライヴ・グラフィック機能も必須のものであり、1990年代前半の初期バージョンからMaxにはリアルタイムのグラフィック処理機能が内包されており、これは現在のMax8でも同等に動作する。任意サイズの「lcd」オブジェクト(いわば白板)にDRAW系の描画を行ったり、静止画を座標指定しつつマルチレイヤー表示してその座標を変化させることでAfterEffectのようなアニメーションまでを実装していた。筆者は1993年頃にはこの機能を使ってカオス描画を実装した[6]が、当時のMacの能力ではその描画スピードには限界があった。

その後、当初はMIDI音楽演奏情報のプログラミング環境としてスタートしたMaxはMacの性能向上に伴ってリアルタイム・サウンド処理機能MSPが加わることになり、同様に外部の「nato」や「Image/ine」ソフトウェアで行っていたライヴ・グラフィクス機能についても、Max/MSPと統合されたjitterという機能によって実装され、jitterはCycing'74社の開発担当者: Joshua Kit Claytonが2002年に筆者のSUAC(静岡文化芸術大学)で開催した「DSPSS2002」において国内から参加

した多数の専門家に対して、世界に先駆けてお披露目された[40]。jitterとは正確にはグラフィクスに特化したものではなく、Max/MSPにおいて最大64次元までのベクトル演算をリアルタイム処理するという機能であり、例えば「1024*768の画素ごとにRGBAの4データを持つベクトル」として画素単位のリアルタイム・グラフィック処理を「ベストエフォート」(コンピュータの処理能力に応じて忙しい場合にはこっそり裏で処理速度を間引いて)実行する、というものである。

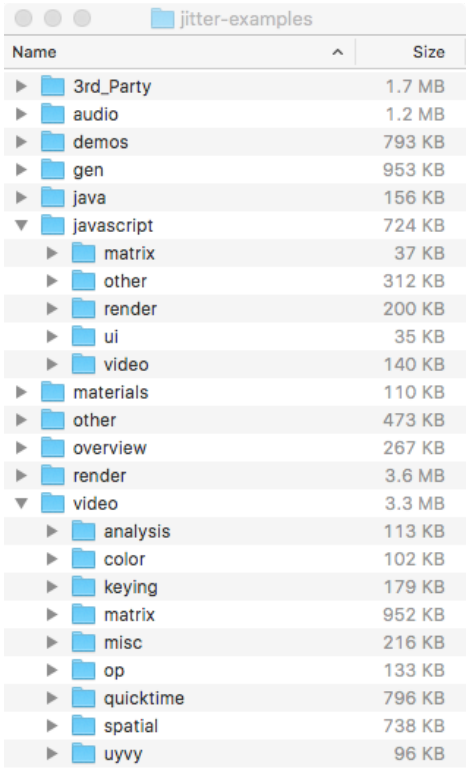

Fig.1 jitter-examples

図1は現在のMax8の「examples」の中にある「jitter-examples」というjitter標準機能の例であるが、このディレクトリには620個のexampleファイルが格納されている。たまたまjavascriptとvideoという2つのカテゴリだけ下の階層まで広げているが、ざっと解説しておくと、まず3rd_Partyとdemosとmaterialsとotherとoverviewは枝葉末節カテゴリなので最初はパスしても良い。audioというのはMSPもあるのに意外な感じがあるが、メモリにバイト単位で直接アクセスするpeek~/poke~やFFT処理などの高速化を実装したカテゴリである。javaとjavascriptはその名の通り、言語記述的なプログラミング環境と橋渡しするカテゴリであり、特にjavascriptにはrender(Open-GL)やvideo(ライヴvideo処理)がほぼ網羅されているが本稿では深入りしない。videoというカテゴリが最も活用されているjitter機能であり、Macの場合にはシステムのQuickTime環境と連携していて、画素単位のanalysis(分析)、色変換処理color、クロマキー処理keying、画素単位のありとありゆる演算処理op、そしてライヴvideo処理のquicktimeなどが網羅されている。ここまでで飛ばしたカテゴリがgenとrenderという2つであり、この重要な2カテゴリについて次節以降で詳しく紹介する。

3.jitterのrenderカテゴリ

筆者が個人所有のSGI(Silicon Graphics社)のIndyワークステーション上でC言語によってOpen-GLで「リアルタイム3D-CG」プログラム(MIDI入力によって3次元空間内の物体が変形/移動したり色彩変化する)を作ったのは、1990年代半ばに「各種言語によるプログラミング例」として実際に過去に開発し動いていた各種言語のサンプルを公開[41]した中のいちばん下にある「IndyのC その7 MIDI制御3次元CG」の頃である。そして、上述のMaxのjitterの各種カテゴリの中のrenderというのは、簡単に言えばこのOpen-GLあたりの概念をほぼ包括したライヴ・グラフィクス機能の一群であり、図2のように厖大なexamplesが提供されている。

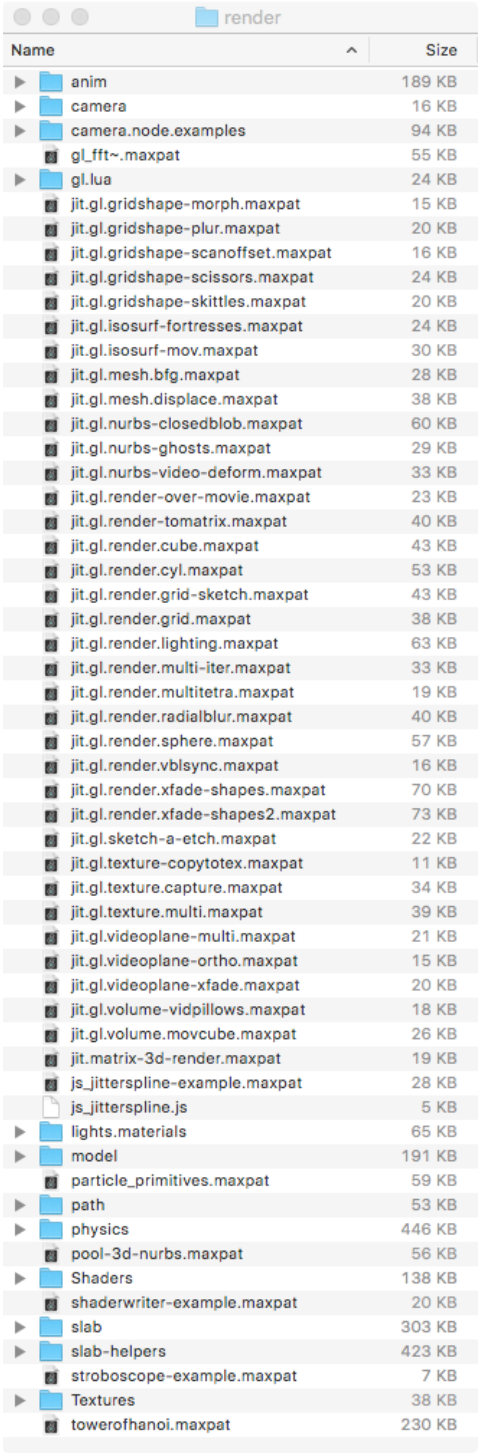

Fig.2 jitter:render

筆者がライヴComputer Music作品の公演においてjitterを活用したのは、Youtube公演記録集[42]によれば2003年にMontrealで初演した作品[43]からであるが、ここで使用していた機能は"video"カテゴリの「movieを可変速再生」・

「lit.lcd上でのDRAW系描画」・「画素単位のエフェクト」等であり、2005年にBancouverで再演した作品[44]や2007年にNewYorkで再演した作品[45]と同様に、まだOpen-GL機能であるrenderカテゴリは未使用であった。

図2の中でディレクトリに入っていないsampleパッチの大部分はOpen-GLで定番となっている機能のサンプルであり、さらにanimは一般の3D-CG/CADシステムの座標系と互換性を持つアニメーション、cameraは3D空間内のカメラでのキャプチャリング、「gl.lua」は南米発の厖大なプログラミング処理系Lua[46]を組み込んだグラフィックライブラリ、lights.materialsは3D空間内のライティング関係、modelは3D空間内で活用するモデリング関係、physicsは物理モデルでの空間記述、Shadersは各種のシェーダ関係、slabはGPUを活用したデータ処理関係などとなっている。筆者がライヴComputer Music作品の公演においてjitterのOpen-GL機能(renderカテゴリ)を活用したのは記録集[42]によれば、2010年にYekaterinburg(Russia)で初演し翌年にOsloで再演した作品[47]や2012年の作品[48]や2016年にBordeaxで初演した作品[49]などであり、本稿執筆時点で作曲中の新作でも後述のGenとともに活用している。

4.Open-GLとopenFrameworks

リアルタイムComputer Graphicsの領域でここ10数年注目されているのがCV(Computer Vision)、すなわちリアルタイム画像認識関係であり、筆者もトヨタ中央研究所からの委託で進めた自動運転車に関する研究[50-51]で関与したが、Max/jitterはそれ以前に開発された体系のためにCVに特化したカテゴリのexamplesは無かった。しかしIAMASのJean-Marc Pelletier氏が公開した「cv.jit」[52]がこれを補完するように主だったCV機能を実現している。

筆者が依頼されてエンタテインメント・コンピューテテングの国際会議ICEC2009(Paris)で行ったTutorial[53]に受講者として参加したロシアのプログラマ/研究者のDenis Perevalov氏との交流は、2010年にロシアで開催された国際電子音響フェスティバル[54]で筆者が行った3件のTutorial[55-57]、さらに2016年の筆者の欧露ツアー[58]におけるYekaterinburg/Moscowでの6件のTutorial/Workshop[59-64]で発展したが、同氏がマルチメディア・プログラミングで活用していたのがopenFrameworks[65-66]である。openFrameworksは単にリアルタイム・グラフィクスのライブラリ集ということではなく、Open-GLを含む各種のグラフィック・ライブラリ、さらにオーディオ(rtAudio等)、フォント(FreeType)、イメージ処理(FreeImage)、動画処理などを含むオープンソースの巨大な体系であるが、商用環境のUnityと競合する状況が続いている。筆者はあくまでプログラミング環境としてのMaxの優位性を優先するので、試用してみた事はあるものの、C++ベースのopenFrameworksよりはむしろ「Processing推し」である。Processing[67]はリアルタイム音響生成システムSuperCollider[68]と共に、OSC(opensoundcontrol)[69]を介してMaxとの相性がとても良好であるのがその理由である。

5.GEN

ここまでを伏線として、ようやく本稿の主題であるGENに到達する。GENとはMax/MSP/jitterの歴史において、コンピュータの性能向上に加えてさらにリアルタイム音響/画像処理に関して飛躍的に性能向上を目指すものとして新たに登場した概念であり、従来からのMaxに対して完全な上位互換性を持っている[70]。Genはこれまでの Maxの特長(箱を並べてラインで繋ぐ)を継承しつつ、Computer Musicにおけるもう一つの大きな柱「言語で記述するプログラミング」までも包含するのが特徴である。Gen自身がMaxの新しいパッチのスタイルをとっており、MSPもjitterも包含していて、Max Gen objectを「gen」[71]と呼び、MSP Gen objectを「gen~」[72]と呼び、Jitter Gen objectsを「jit.gen, jit.pix, jit.gl.pix」[73]と呼ぶ。GenパッチはGenオブジェクトの中身の処理を記述している。その処理は図3のように内部的には GenExpr として言語的に記述され、コンパイルされている。

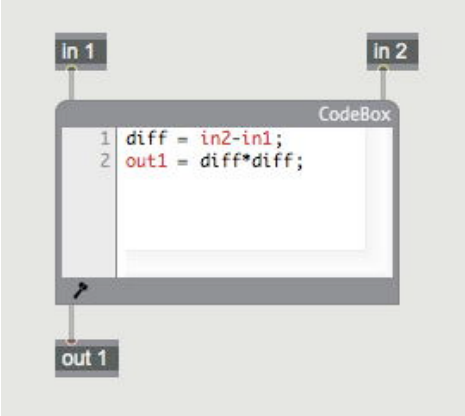

Fig.3 Gen patch

図4は、図1のMax8の「examples」の中にある「jitter-examples」の中の「gen」ディレクトリの中身であるが、同じ名前で1から3まで連番のパッチの2を二つ除外することでようやくスクリーンショットが撮れたほど多数の中身として充実している。Gen Overviewというページ[70]に続いてその名もGenという解説ページ[74]を見ると、以下のような興味をそそられる豊富なリンクが並んでいた。これらのGen解説ドキュメント類は一斉に整えられたものではなく、Maxのオープンソースコミュニティにおいて逐次的に積み上げられてきた知的成果であり、簡単な解説の重複があったり著者ごとのクセがあったりするものの、全体を読破してみるとGenの概要は容易に理解できる。

Gen

The gen~ for Beginners Series

- gen~ for Beginners, Part 1: A Place to Start
- gen~ for Beginners, Part 2: Similarities and Differences
- gen~ for Beginners, Part 3: Counting, and a World without bang Messages
- gen~ for Beginners, Part 4: Working with buffers (and data)
- gen~ for Beginners, Part 5: The codebox Operator
- gen~ for Beginners, Part 6: Thinking Inside the Codebox
- gen~ for Beginners, Part 7: Creating Reusable Tools

Online Tutorials

- Gen Tutorial 1 - The Garden of Earthly Delays
- Gen Tutorial 2a - The Joy of Swiz
- Gen Tutorial 2b - Adventures in Vectorland
- Gen Tutorial 3 - The Fine Art of Surfacing

Gen Patcher Collections

- Gen Patch-a-Day Tutorials

gen~ Example Patches

- (ここに多数のgen~ Example Patchesが並んでいる)

Gen Code Export Tutorials

- Code Export for Audio Units
- Code Export for VST
- Code Export for iOS
- Code Export for ISF

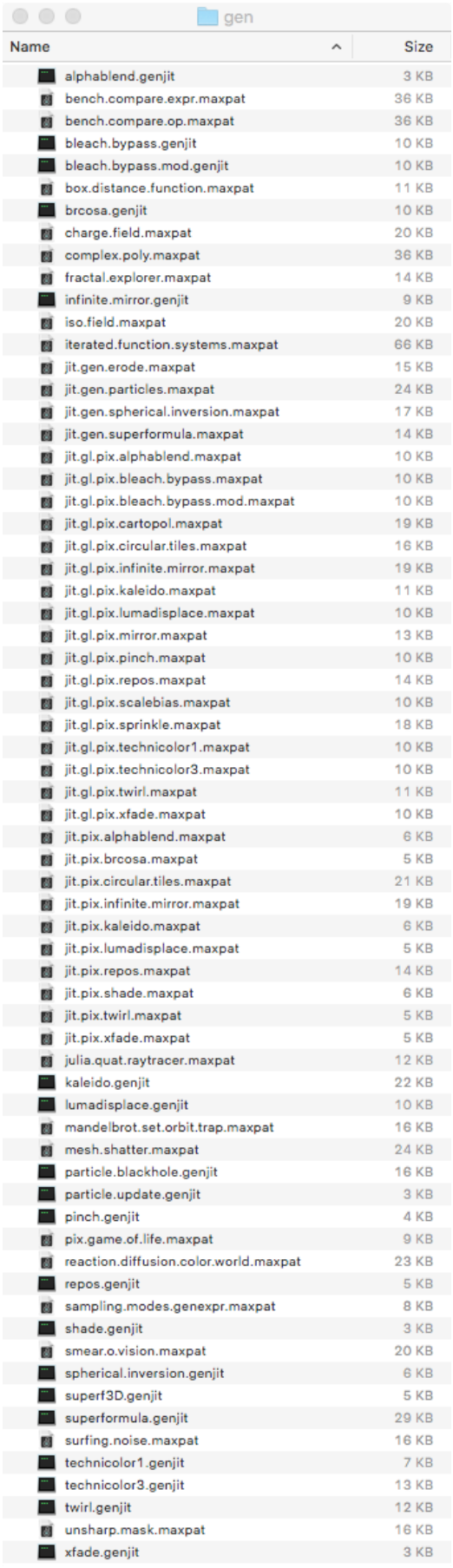

Fig. 4 jitter:gen

Maxパッチの左下端にあるボタン(宝箱アイコンの蓋が開閉トグル動作)は、パッチ内にオブジェクトを配置したりワイヤリングする「編集モード」(unlock)と実際に動作する「実行モード」(lock)を行き来するもので、ユーザから見るとGUIベースのインタプリタのように扱えるが、lockした瞬間に内部的には最適化/高速化するコンパイラが自動実行されている。「gen」[71]ではこれをさらに高度化して、単純な機能を非常に多数並べたような場合には最適化コンパイルによって相当に高速化される。また「gen~」[72]が元々GEN開発の原動力となったものであり、世界中の音楽音響信号処理に関するアルゴリズムを網羅したMusicdsp[75]にあるような、各種の言語形式で記述された古今東西の信号処理アルゴリズムをMSPの枠組みにシームレスに組み込むために提供されてきた。これには多数の専門家(Maxの熱心な支援者)がおよそ10年間がかりで関与してきたために、その実装形態には色々な形式があり(同じ処理でも何パターンもの実現方法あり)、筆者が調べた経緯を[76]で公開しているが、その全貌は相当な知的財産と言える。

6.Jitter Gen objects

そして本稿のテーマはMaxのライヴ・グラフィクス機能をGEN化した「Jitter Gen objects」[73]であり、これには「jit.gen」・「jit.pix」・「jit.gl.pix」の3種類がある。前2つは上述の「gen」と「gen~」と同様に透過的にネイティブCPUマシンコードをオンザフライで生成してコンパイルし、「jit.gl.pix」はOpen-GLなのでGPUコード(GLSL)に対して透過的にネイティブマシンコードをオンザフライで生成してコンパイルする。「jit.pix」と「jit.gl.pix」オブジェクトは2次元イメージ処理用に限定されていて、1～4プレーン(RGBA)の入力マトリックスで動作し、入力されたデータをイメージデータのように扱う。そして「jit.gen」オブジェクトは、マトリックスを含むあらゆるタイプのデータを扱う一般的なjitterマトリックス処理オブジェクトである。

Fig. 4 元画像

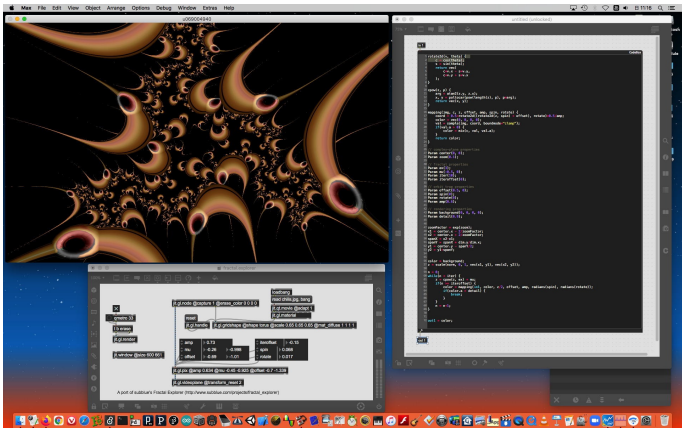

Fig. 5 生成中のライヴグラフィクス例

本稿では紙面と学会提出PDFサイズに制限があるため、ここでは[76]の調査の中で、Genサンプル集[74]の中にあった、その名も「fractal.explorer.maxpat」について紹介する。このサンプルでは、図4のような元画像[77]を材料として使用して、図5のようなフラクタル画像(高解像度版は[78])をライヴ生成している。これまでの「フラクタル画像」(静止画)との違いは、このウインドウ(フルスクリーンに展開可能)内のグラフィックは全画素に関してライヴ生成され続けていることであり、Genボックス内に記述されているパラメータをMaxパッチから刻々と変化させてみると、グラフィック全体のフラクタル構造(細部は無限小まで繰り返されている)がうにょうにょ・・・と変化する、という点にある。

Maxが30年前から持っているGUI(スライダーやダイヤル)でこれらGenパラメータを変化させてみると、マウスの操作に対応してフラクタル画像がサワサワと動き、MSPのマイク入力音圧をGenパラメータに割り当てれば、パソコンに声をかけるとそれに反応して無限小まで続くフラクタルが生き物のように反応する。筆者はComputer Musicの作曲の一部として多種のセンサを「新楽器」としてきたが、例えば多チャンネル「テルミン」(距離センサ群)の前で手をひらひらすればフラクタルがわらわらと動き、筋電センサなど生体情報センサに繋げば身体状況/精神状況に対応してフラクタルがライヴに変化する。当然のことながらこれらライヴ・パラメータに基づいてGen生成のライヴ音響も生まれているので、聴衆(体験者)はもちろん演奏者(Perfoemer)自身がこのインタラクティブ・マルチメディア関係性をバイオフィードバックとして体験し、筆者が追求している「ウェルネス・エンタテインメント」の世界に限りなく近付いていく事を実感しているところである。

7.RNBO(速報)

Maxは30年間以上も進化を続けてきているが、現在でもさらに進化を進めている。本稿執筆時点の2022年11月2日にMax提供元のCycling’74社から届いた案内は、それまでの安定版Max8.3.3から一気にMax8.5にスキップした上で新たに搭載されたRNBO[79]という機能の紹介であった。GUIベースの柔軟なプログラミング環境であるMaxは実行時には内部的にコンパイルして高速化し、それをさらに特化させたものがGENであったが、つまりMaxは「高機能コンパイラの塊」であるとも言える。これを発展させて、「Maxで開発したコードを最適化コンパイルして外部の機器にロードする」というのがRNBOであり、Maxで開発したMSPベースのリアルタイム音響信号処理プログラムをiPhoneやRaspberry PiやAudioプラグインやWebブラウザ内で走らせることが容易になった。詳細についてはいずれ機会があれば報告したい。

8.おわりに

マンデルブロ集合やジュリア集合の美しいグラフィクスは、誰に対してもフラクタルやカオスの「数理造形」の美を圧倒的な迫力で伝えてくれるが、印刷された紙面だったり静止画の画像だったりして、それを作成するためのコンピューティング労力は実際に「描画プログラム」と取り組んでみた者だけが体感していた。これを活用したいメディアアートの世界では、フラクタル構造の微細化の深淵まで計算させることの難しさのため、事前にレンダリング制作した静止画を多数用意したアニメーション(パラパラ動画)で誤魔化すしか方法が無かった時代が長く続いたが、コンピュータの性能向上によって、ようやく「ライヴでフラクタル描画を刻々と生成」という領域に到達してきた。本稿で紹介したGENはその一領域を担って多くの作家/専門家を刺激しており、数理的現象のvisualizationという意味で、アートだけでなく情報科学/非線形科学の領域でも意義を持つ可能性があり、今後も研究を進めていきたい。

文　献

1. 長嶋洋一. Chaotic Interaction Model for Hierarchical Structure in Music. 平成5年度前期全国大会講演論文集II, 情報処理学会, 1993
2. 長嶋洋一. Musical Concept and System Design of "Chaotic Grains". 情報処理学会研究報告 Vol.93,No.32 (93-MUS-1), 情報処理学会, 1993
3. 長嶋洋一. Chaotic Interaction Model for Real-Time Composition. 人工知能学会全国大会論文集I, 人工知能学会, 1993
4. Yoichi Nagashima. PEGASUS-2 : Real-Time Composing Environment with Chaotic Interaction Model, Proceedings of 1993 International Computer Music Conference, ICMA, 1993
5. Yoichi Nagashima. Chaotic Interaction Model for Compositional Structure, Proceedings of IAKTA / LIST International Workshop on Knowledge Technology in the Arts, 1993
6. 長嶋洋一. Chaos理論とComputer Music. 京都芸術短期大学紀要［瓜生］第16号, 京都芸術短期大学, 1994
7. 長嶋洋一. Attractor Synthesisによる楽音合成システムの検討. 平成6年度前期全国大会講演論文集I, 情報処理学会, 1994
8. 長嶋洋一. マルチメディア作品におけるカオス情報処理の応用(研究ノート). 京都芸術短期大学紀要［瓜生］第18号, 京都芸術短期大学, 1996
9. 長嶋洋一. 非線形科学の視点から「コンピュータ音楽」を考える. 電子情報通信学会非線形問題研究会（NLP）研究会資料（技術研究報告） NLP2010-133, 電子情報通信学会, 2011
10. 長嶋洋一. カオスに対する聴覚的なアプローチ(1). 電子情報通信学会非線形問題研究会（NLP）研究会資料（技術研究報告） NLP2011-158, 電子情報通信学会, 2012
11. 長嶋洋一. サウンド知覚のカオス共鳴によるモデル化に向けて. 電子情報通信学会非線形問題研究会（NLP）研究会資料（技術研究報告） NLP2013-144, 電子情報通信学会, 2014
12. 長嶋洋一. カオスに対する聴覚的なアプローチ(2). 電子情報通信学会非線形問題研究会（NLP）研究会資料（技術研究報告） NLP2014-44, 電子情報通信学会, 2014
13. 長嶋洋一. 内受容感覚とバイオフィードバックに注目した筋電情報ジェスチャ認識によるエンタテインメント. 情報科学技術フォーラム2015講演論文集, 情報処理学会・電子情報通信学会, 2015
14. 長嶋洋一. 内受容感覚バイオフィードバックによる"癒し系エンタテインメント"の考察. エンタテインメントコンピューティング2015論文集, EC2015実行委員会, 2015
15. Yoichi Nagashima. Towards the BioFeedback Game --- with Interoception and Rehabilitation ---. Proceedings of the 8th International Conference on Virtual Worlds and Games for Serious Applications, VS-Games, 2016
16. Yoichi Nagashima. Bio-Sensing and Bio-Feedback Instruments --- DoubleMyo, MuseOSC and MRTI2015 ---. Proceedings of 2016 International Computer Music Conference, ICMA, 2016

17. 長嶋洋一. 生体情報センシングのバイオフィードバック療法への応用について, 知覚情報研究会・研究報告, 電気学会, 2017
18. 長嶋洋一. 生体情報センシングと内受容感覚コミュニケーションの可能性について. 電子情報通信学会ヒューマンコミュニケーション基礎研究会資料（技術研究報告） HCS2017-102, 電子情報通信学会, 2018
19. 長嶋洋一. 触覚バイオフィードバック」汎用プラットフォームの提案 -メディアアートのウェルネスデザイン応用を目指して-. 電子情報通信学会ヒューマン情報処理研究会資料（技術研究報告） HIP2018-39, 電子情報通信学会, 2018
20. 長嶋洋一. ウェルネス・エンターテインメントを実現するメデイアアート. 京都市立芸術大学 美術研究科（メディアアート）博士（後期）課程 博士論文, 2019 https://nagasm.org/ASL/paper/KCUA_nagasm_final.pdf
21. 長嶋洋一. ウェルネス・エンタテインメントのための錯覚体験システム ～ 聴覚やマルチモーダル錯覚を中心として ～. 電子情報通信学会ヒューマン情報処理研究会資料（技術研究報告） HIP2019-87, 電子情報通信学会, 2020
22. 長嶋洋一. メディアデザインにおけるバイオフィードバック応用の事例報告, 電子情報通信学会MEとバイオサイバネティックス研究会資料（技術研究報告） MBE2021-01, 電子情報通信学会, 2021
23. Max前夜 https://nagasm.org/ASL/max02/
24. "Wandering Highlander" Max Patch https://nagasm.org/ASL/wander/patch.html
25. 生体センサとMax4/MSP2による事例報告 https://nagasm.org/ASL/SIGMUS0202/
26. インタラクティブアートの統合的システム・プラットフォームとしてのMax/MSP https://nagasm.org/ASL/dspss2002/
27. Max/MSPとKymaとLabVIEWによる音響処理について https://nagasm.org/ASL/paper/IPSJ0303.pdf
28. 生体センサとMax4/MSP2による事例報告 https://nagasm.org/ASL/SIGMUS0202/
29. Max6の作者自身が作るMaxパッチを見てみよう http://www.youtube.com/watch?v=Kxr0Z7SBUuQ
30. Maxでブラウザを作ろう(jweb) https://nagasm.org/1106/news4/docs/jweb.jpg
31. USBコントローラ用Maxパッチを作る https://nagasm.org/1106/news5/20170419/
32. Max6日記 https://nagasm.org/ASL/max03/
33. Max7日記 https://nagasm.org/ASL/Max7_1/
34. 続Max7日記 https://nagasm.org/ASL/Max7_part2_1/
35. 基礎心理学実験プロトタイピングツールとしてのMax7とウェルネスエンタテインメントプラットフォームとしてのMax7 https://nagasm.org/ASL/paper/SIGMUS201808.pdf
36. 音楽情報科学ツール"Max"を用いたメディアデザイン -RFIDの活用例を中心として https://nagasm.org/ASL/paper/sigmus201908-1.pdf
37. 長嶋洋一. アルゴリズム作曲. コンピュータと音楽の世界, 共立出版, 1998
38. https://nagasm.org/ASL/max01/
39. https://www.ircam.fr/#
40. https://nagasm.org/1106/news5/docs/DSPSS2002.html
41. https://nagasm.org/ASL/program/
42. https://nagasm.org/ASL/YouTube.html
43. https://www.youtube.com/watch?v=H8-AeibByWI
44. https://www.youtube.com/watch?v=Rd-mPax3hS8
45. https://www.youtube.com/watch?v=lcoANcRQ0ao
46. http://www.lua.org/
47. https://www.youtube.com/watch?v=32FLFkgZYKk
48. http://www.youtube.com/watch?v=GiO0hVnoA0c
49. https://nagasm.org/Sabbatical2016/tempora2016_rehearsal.mp4
50. 長嶋洋一. 自動運転車のためのリアルタイム作曲システムに向けて. 情報処理学会研究報告(2017-MUS-118), 情報処理学会, 2018
51. Yoichi Nagashima. Realtime Musical Composition System for Automatic Driving Vehicles. Proceedings of 2018 International Conference on Entertainment Computing (IFIP TC14 ICEC) , 2018
52. https://jmpelletier.com/cvjit/
53. https://nagasm.org/ASL/ICEC2009/
54. https://nagasm.org/1106/SYNC2010/
55. https://nagasm.org/ASL/SYNC2010_Lecture_1/
56. https://nagasm.org/ASL/SYNC2010_Lecture_2/
57. https://nagasm.org/ASL/SYNC2010_Lecture_3/
58. https://nagasm.org/1106/news5/docs/Tour2016.html
59. https://nagasm.org/1106/news5/Russia_Lecture_1/
60. https://nagasm.org/1106/news5/Russia_Lecture_2/
61. https://nagasm.org/1106/news5/Russia_Workshop_1/
62. https://nagasm.org/1106/news5/Russia_Workshop_2/
63. https://nagasm.org/1106/news5/Russia_Lecture_3/
64. https://nagasm.org/1106/news5/Russia_Lecture_4/
65. https://www.packtpub.com/product/mastering-openframeworks-creative-coding-demystified/9781849518048
66. https://www.packtpub.com/product/openframeworks-essentials/9781784396145
67. https://processing.org/
68. https://supercollider.github.io/
69. https://opensoundcontrol.stanford.edu/
70. https://docs.cycling74.com/max8/vignettes/gen_overview
71. https://docs.cycling74.com/max8/vignettes/gen_common_operators
72. https://docs.cycling74.com/max8/vignettes/gen~_operators
73. https://docs.cycling74.com/max8/vignettes/gen_jitter_operators
74. https://docs.cycling74.com/max8/vignettes/gen_topic
75. https://www.musicdsp.org/
76. https://nagasm.org/ASL/Sketch14/
77. https://nagasm.org/ASL/Sketch14/fig7/010.jpg
78. https://nagasm.org/ASL/Sketch14/fig7/009.jpg
79. https://cycling74.com/products/rnbo