

Risset Rhythm 温故知新

Learning from the past : Risset Rhythm

長嶋 洋一

Yoichi Nagashima

静岡文化芸術大学

Shizuoka University of Art and Culture

nagasm@suac.ac.jp

内容梗概：ピッチに関する錯聴のShepard Toneを、Jean-Claude Rissetがリズム/テンポに拡張した錯聴がRisset Rhythmである。本稿ではインタラクティブな錯覚体験を指向するメディアデザイン教育に関連して、これら2種の錯聴の実験/実装を報告した第1報に続いて、ドラム音を中心としたリズムパターンでなく音楽断片を切り出したループ音楽のRisset Rhythm化についての実験、「上昇しつつ下降する」Shepard Toneの実験等について紹介するとともに、音楽におけるRisset Rhythmの新たな可能性について検討してみたい。

Keywords: Auditory Illusion, Shepard Tone, Risset Rhythm

1. はじめに

ピッチに関する有名な錯聴のShepard Toneを、Jean-Claude Rissetがリズムに拡張した錯聴がRisset Rhythmであり、音楽情報科学する者であればいずれも教養として知っていた古典的な話題である。メディアデザイン教育において錯覚をテーマで取り上げると、必ずと言っていいほどShepard Toneは登場するものの、Risset Rhythmについては本人監修?のデモYouTube[1]のサウンド[2]を鑑賞する程度で過ぎ去ってしまうことも多い。本稿では、錯覚を鑑賞の対象としてたんに再生するだけでなく、インタラクティブな錯覚体験を指向するメディアデザイン教育に関連して、COVID-19のために遠隔教育体制となった2020年前期にあらためてRisset Rhythmを温故知新として取り上げた事例を紹介する。この発端は、Risset Rhythmについて、筆者のゼミに北京から遠隔ZOOMで参加しているSUAC研究生(王福瀛)が研究テーマとして探してきたことを発端に、筆者のSUAC(静岡文化芸術大学)での専門科目「音楽情報科学」の受講学生とともに検討するための資料を作ってみよう、と思い立ったことである。当初のはRisset Rhythmの原理を理解した上で、このサウンドを生成するMaxパッチ、あるいは手法を確立することを目標としたが、音楽におけるテンポは「情動」に深く関係することから、音楽におけるRisset Rhythmの新たな可能性についても検討したいと考えている。

2. 前半部分の概要

本研究の第1報として、情報処理学会音楽情報科学研究会・夏のシンポジウム(2020年8月・

オンライン)によって報告した[3]ので、この前半部分を概説する。まず第1章「Shepard Toneの復習」として、純音[4]および鋸歯状波[5]によるShepard Tone生成Maxパッチを作成した。ここでは聴覚的に等速度で連続的なピッチ変化(上昇/下降)を実現するために、線形増減変化オブジェクト「line」から与えられた浮動小数点数値を「mtof」(MIDIノートナンバーから基音の周波数に変換)に与えるというあまり知られていないノウハウを活用した。

続く第2章「Ghost Notes仮説」では、Risset Rhythmの原理はShepard Toneと同様に「複合音の漸増/漸減」に対して聴覚上の注意が騙されると想定して、スネアドラム等で細かく裏箔で刻む「ゴーストノート」がテンポの加速とともに浮上してくるという仮説に基づいたMaxパッチを作成した。ここでは聴覚的に等速度で連続的なテンポ変化(加速/減速)を実現するために、一定速度のリズムパターンのサウンドファイルを作り、これを可変ピッチで再生する「groove~」の読み出しピッチ情報として「line」の線形増加数値を指数関数変換して与えた。このサウンド[6]については、騒然としているものの、よく聞いてみるとある程度「次第に加速している」感じもあるという学生の感想を得た。

続く第3章「文献調査」では、行き詰まった仮説の改良のためにWebで文献などを調査した。そしてShepard Toneでは考慮が不要なのに対してRisset Rhythmでは「位相が重要」との記述に触発されて、新たに第4章「Loopからの生成とCrossFade」へと展開した。「groove~」がループ再生終了時に出力する情報を活用することで、オクターブ単位の速度変化ブロックごとに同期するように実装してみたが、ブ

ロック端点の設計に不備があり、生成されるサウンド[7]に不満が残った。そして再び文献検討した際に確認した図[8]にトリガされて、**第5章「2の累乗の完全Loop」**という解決編に到達した。このパッチ[9]は全体が5つの加速ブロックからなっていて、下から順に「2オクターブと6半音下→1オクターブと6半音下→6半音下」に加速する第1ブロック、「1オクターブと6半音下→6半音上」に加速する第2ブロック、「6半音上→1オクターブと6半音上」に加速する第3ブロック、「1オクターブと6半音上→2オクターブと6半音上」に加速する第4ブロック、「1オクターブと6半音上→2オクターブと6半音上」に加速する第5ブロック、からなる。この各ブロック(metabarという)の継ぎ目が連結することで全体は「**加速ブロックの螺旋**」のような構造となり、自然に連続して無限に加速し続けるサウンドが実現される。これがRisset Rhythmの本質であった。

この理解のもとに、**第6章「別サウンドで手法の確認」**として、ピッチ感のある音を含むリズムについても良好なRisset Rhythm[10]が生成でき、さらに**第7章「逆方向:無限に遅くなるサウンド」**として、単純にMaxパッチの各ブロックでの指定として始点と終点を入れ替えるで、良好な「無限に減速する」Risset Rhythm[11][12]が生成できた。

文献[13]の最後のあたりで、Risset Rhythmの音楽的な考察をしていた。これまでの作曲理論や音楽美学では、音楽の速度がテンポアップすることは、気持ちの高揚や興奮といった影響があり、作曲家も指揮者も演奏家も聴衆も、その効果を暗黙の共通理解として共有していた。ところがRisset Rhythmで「無限に加速する」となれば、そんなに永遠に気持ちの高揚や興奮が続くはずがない(死んでしまう)ので、これは音楽美学的に面白い問題提起であると言える。そして「無限に減速するサウンド」も実現できたが、音楽心理学的には、テンポダウンによって気持ちが落ち着いたり気分が落ち込んだりすることと、「無限に減速する」となれば、どこまでも落ち込む気分を齎す可能性がある。以上ここまでを、音楽情報科学研究会・夏のシンポジウムにおいて前半部分として報告した。

3. We Will Rock You

上述のように原理を理解してRisset Rhythm生成Maxパッチが完成してみると、単純なパターンをループさせてRisset Rhythmにするのは、ほぼ以下同文で出来そうな気がしてきた。そこで思い出したのが、2007年7月の欧州ツアー2007[14]で立ち寄ったロンドンで体験した、QueenのRockミュージカル「We Will Rock You」の、あの感動的なリフレインだった。そこで、メロディーと長さがそこそこあるこの印象的なリフレインを素材としたらどうなるかを実験してみた(**第8章**)。

まず、ずっと昔に購入していたCDから「We Will Rock You」の冒頭のリフレイン(1回分:4小節)を切り出し、手作業でトリミングしてルー

プ素材を作った。そして「LoopChecker」Maxパッチを使い、最低速度ブロックで4小節のリフレイン1回の所要時間(metabar)を23155msecと決めて実験してみたところ、確かにあの曲の面影もあるし、確かに加速していたが、わずかにどこかが物足りない気もした。

この不自然さの原因は、ピッチ換算でオクターブ、つまり最低音域で原曲から倍速までテンポ変化する時間枠(metabar)が23155msecと短いためかと推理し、「LoopChecker」のループ指定を「1」から「2」に、つまり1回分4小節のリフレインを「2回ループ」するように改訂・微調整して探した結果、所要時間(metabar)を46285msecとした。そして改訂したパッチ[15]で作ってみたサウンド[16]では、オクターブ上のループがさきほどの最低音域のスピードで鳴っているので「聴感上のテンポ感」はほとんど同じだったが、注意して聞いてみると、いちばん下の音域から重低音のループが出てくるあたりがわずかに違い、こちらの方がやや好ましい印象があった。

そこでこのパッチ[15]の起点と終点を入れ替えたパッチで作った、「無限に遅くなる「We Will Rock You」」のサウンド[17]を聞いてみると、盛り上がる筈の原曲が、どこまでも落ち込んでいく不思議な気分になった。聴覚的な「注意」が下降していくサウンドに向いているため、いちばん下の音域の重低音のループがあることが有効であるように思われた。

4. サウンド素材特性の実験

上述の実験から、ピッチ感/和声感の無いドラム楽器によるリズムパターンだけでなく、ほぼ同じコードとはいっても和声的要素のある音楽的ループについてもRisset Rhythm化できる、という興味ある知見が得られた。そうになると、ループ内でコードが進行するとか、ループの前半はスカスカで後半に要素が詰まっているとか、さらに音楽的な要素でどうなるか、という興味が生じた。そこで**第9章**として、さらに大きくキャラの異なる3種類のサウンド素材を使つての実験を進めた。素材の歌手としてはせっかくなので筆者と同級生の、マイケルジャクソン(鬼籍入り)と石川さゆり(同年生まれ1学年上)と山口百恵、という3人を選択し、いずれもずっと昔に購入していたCDから取り出した。実験で得られたサウンド(ループ1回/2回ないし2回/4回)は以下である。

素材[A] : マイケルジャクソン 「スリラー」より

「metabar=ループ1回」の実験 : metabar = 31842msec
無限加速 : [18]
無限減速 : [19]

「metabar=ループ2回」の実験 : metabar = 63674msec

無限加速 : [20]
無限減速 : [21]

素材[B] : 石川さゆり 「津軽海峡・冬景色」より

「metabar=ループ1回」の実験 : metabar = 46260msec
無限加速 : [22]
無限減速 : [23]

「metabar=ループ2回」の実験：metabar = 92520msec
無限加速：[24]
無限減速：[25]

素材[C]：山口百恵「プレイバックpart2」より

「metabar=ループ2回」の実験：metabar = 30150msec
無限加速：[26]
無限減速：[27]

「metabar=ループ4回」の実験：metabar = 60290msec
無限加速：[28]
無限減速：[29]

実験では「1/2回ループ」(metabarが短い時間)と「2/4回ループ」との比較に注目して聞いてみたが、まず連続加速版ではその違いは感じられなかった。そして連続減速版において、最低速度音域の動きに注目してみたが、metabarが60秒という領域は十分すぎる低音域のためか、ほとんど違いは感じられなかった。ここまでの経験則(概略)としては、最初のRissetオリジナルRisset Rhythmのmetabarが約30秒であったが、オーダとしてその倍の60秒というのはおそらくoverspecというか無駄、という感じであり、唯一、「metabar=23秒」というサンプルでの連続減速版で、わずかに聞き分けられるような違いがあった。最初はループの先頭にビートがある「打楽器系」のサウンドがRisset Rhythmに適しているという先入観があったが、ずっと鳴っている素材[A]や素材[B]でも、フレーズの後半に重点のある素材[C]でも、それなりの面白いサウンドが得られたことは大きな収穫である。

5. 「逆Shepard Tone」の復習

ループ音楽の素材としてコード進行や複雑な要素を持っていてもRisset Rhythm化できたところで、実験はちょっとだけ回り道をして「Shepard Toneの復習」に戻った。だいぶ昔(1990年代前半)にShepard Toneを作るためにワークステーションで苦労してやったことも今ならMaxで簡単だと確認できたので、第10章として、かつて苦労して作った不思議なサウンドも作ってみたいとなったからである。まず材料として64秒間の無限に上昇するShepard Toneを作り、次にこれを波形編集ソフトに読み込んで、「リバーブ」つまり逆方向に時間反転して書き出した、64秒間の無限に下降するShepard Toneを作った。

そしてMaxパッチ[30]は、「無限に上昇するサウンド」を再生している期間に、バンドパスフィルタの中心周波数が指数関数特性で最高音域から最低音域まで「下降」するものである。その結果、得られるサウンドは、内部的には「上がり続けている」のに、全体としては高音域から低音域に「下がり続けている」ように知覚されるサウンド[31]となる。同様に、「無限に下降するサウンド」を再生している期間に、バンドパスフィルタの中心周波数が指数関数特性で最低音域から最高音域まで「上昇」するというパッチから得られるサウンドは、内部的には「下がり続けている」

のに、全体としては低音域から高音域に「上がり続けている」ように知覚されるサウンド[32]となる。これがかつて筆者は「逆Shepard Tone」と呼んだかもしれないが、ここで問題が提起された。Risset Rhythmで、これは出来るだろうか、出来ないだろうか。つまり、「どんどん加速しつつどんどん減速するリズム/サウンド」または「どんどん減速しつつどんどん加速するリズム/サウンド」というのはあるのか？ もし、これが出来れば、「逆Risset Rhythm」と言えるだろう。

6. 新たな「逆Shepard Tone」探求

COVID-19環境下のために常に問題意識を持って日々を過ごす、というのはいい事でもあった。ある時、本命の「逆Risset Rhythm」の前に、まずはShepard Toneで実験すべきアイデアが「3つ」と浮かんできたので、そこから実験に着手した。既に開発したRisset Rhythmの生成パッチは、スタートした瞬間から「無限に加速/減速する」サウンドを生成し続けるが、第1章のShepard Tone生成パッチは、いくつものパーシャルが順に起動していった一巡するとうまく「無限に上昇/下降する」サウンドを生成し、それを「record」で保存したサウンドファイルを後で「再生」するので、その長さが有限(ちょっとした未完成)だった。そこでまずスタートラインとして、「無限に上昇/下降する」サウンドを生成し続ける、という基準Shepard Tone音源パッチを用意した。

パッチ[33]は、素材として「30秒間の上昇Shepard Tone」ファイルを使って、上段では3つのブロックがこれをそのまま再生していて、「10秒間でゼロ→1に音量増加」「10秒間はそのままの音量」「10秒間で1→ゼロに音量現象」という動作を10秒ごとに繰り返すことで、「無限に上昇する」サウンドを生成し続ける。下段では3つのブロックが素材ファイルを逆方向に再生していて「10秒間でゼロ→1に音量増加」「10秒間はそのままの音量」「10秒間で1→ゼロに音量現象」という動作を10秒ごとに繰り返すことで、「無限に下降する」サウンドを生成し続ける。そしてこのパッチ[33]では、その2系統の生成サウンドをモノラルのまま足しあわせているので、結果として得られるサウンドは上昇も下降もなく、適当に短い上下を繰り返したようなサウンド[34]になった。上段と下段のスタートの時間差「delay 123」(123msecの遅延)は、他に10とか100とか1000とか適当にしてもほぼ響きは同様である。実験すべきアイデアの「第1」のこれが駄目だろうというのは、最初から想定内であった。

実験したいと思ったアイデアの「第2」がパッチ[35]での実験であり、Shepard Toneの生成部分は[33]とまったく同一である。パッチ[33]との違いは、「無限に上昇」と「無限に下降」のそれぞれの系列を、ステレオの左右に分離したということである。ここはスピーカでなくイヤホンかヘッドホンで聞かないと絶対に無意味となるが、サウンド[36]のようになっ

た。そのつもりで聞いてみれば、左右それぞれのサウンドは「無限に上昇」と「無限に下降」となりそうだが、サウンド[34]のミックスと似たような繰り返しの周期が感じられるので、**脳内で左右のチャンネルのサウンドがミックスされる機構が働いている模様**であり、聴覚心理学実験として新たにやってみなければならぬテーマかもしれない。

実験したいと思ったアイデアの「第3」がパッチ[37]での実験であり、Shepard Toneの生成部分はまったく同一である。ここでは出力サウンドは60秒間の有限なファイルとなり、サウンド[38]になった。これは「第2」の実験で左右に分かれていたチャンネルのサウンドを、60秒間かけて、片方はゆっくりと左→右に、もう片方はゆっくりと右→左に**音像を移動**させる、というPANPOTの実験である。過去に電子音響音楽の作曲において、全てのサウンドの定位を固定させず、例えば「常にゆっくりと左→右に移動」などのルールを適用すると、それぞれの音源が混濁せずに浮き立ってくれた、という経験則がここで生きないか・・・という仮説だったが、結果は無残にも駄目だった。それぞれのサウンドが移動して、中央付近では「第1」の実験のようにミックスされて不自然な上下フレーズようになってShepard感が消えてしまい、ちょっと期待はずれという結果になった(第11章)。

7. metabar生成規則性の探求

前章の実験から、比較的シンプルなShepard Toneであっても「上昇しつつ下降」「下降しつつ上昇」というのは「パンポット応用」の作戦では難しいと判明したのでちょっと棚上げし、手作業のトライアンドエラーで進めてきたRisset Rhythm生成アルゴリズムの「metabar」計測について、ここまでの事例から法則性を見出せるかどうかを整理した。完全なRisset Rhythmが生成できるようになった第5章から第9章までのMaxパッチに関して、「ループ素材の長さ」と「最低速音域のループ数」と「metabarの長さ」との間に、例えば簡単な線形性などの数学的関係性が見出せるかという検討である。数学的関係性があれば、今後は任意のループ素材に対して「最低速音域のループ数」だけ指定すると簡単にRisset Rhythmが生成できることになり、ライブComputer Musicの場においては、ステージ上でライブサンプリングした音素材をその場で次々にRisset Rhythm化するという、これまで世界で誰もやってこなかった新しい音楽の出現となる可能性がある。インタラクティブComputer Music派の筆者としては真正面から取り組みたい大テーマとなる。

そしてうすうすそんな感じはしていたが、図1のように電卓で計算して整理してみると、綺麗に比例関係が見出せた。LoopCheckerのパッチでは「groove~」のループエンド検出部分にレバトリガの本質的な誤差が生じていたが、図1の表はまずまず誤差を含めた上でも良くで

きている。おそらく数学的には時間的に積分してちゃんと出てくると思うのだが、「[b]/[a]の比」については、およそ3.9218あたり(9件の平均は3.9217855)で、あとは誤差という感じである。これにより逆に厳密にキチンとした長さのサンプル音素材を用意して、そこからRisset Rhythmを自動生成するパッチを作る、という道が見えてきた。第9章で考察したように、metabarの長さのオーダーとしては、「オリジナルRisset Rhythm」が「metabar=約30秒」であり、オーダーとしてその倍の60秒というのはoverspecというかおそらく無駄である。また、唯一「metabar=23秒」というサンプルでの連続減速版で、わずかに聞き分けられるような違いがあったことから、目安として「metabar=約25秒以上」(50秒未満)という範囲がここまでの経験則となる。

章	ループチェック	ループ素材	ループ素材の長さ	ループ回数	素材×回数の長さ [a]	metabarの長さ [b]	[b] / [a] の比
第5章	LoopChecker.maxpat	break3.aif	1846.666633	4	7386.6664	28962	3.9208485
第6章	LoopChecker2.maxpat	beat1.aif	4075.011263	2	8150.0224	31952	3.92048
第8章	LoopChecker3.maxpat	RockYou4.aif	5901.677896	1	5901.677896	23155	3.9234605
第8章	LoopChecker4.maxpat	RockYou4.aif	5901.677896	2	11803.355	46285	3.9213427
第9章	LoopChecker5.maxpat	MJ04.aif	8118.344522	1	8118.344522	31842	3.922282
第9章	LoopChecker6.maxpat	MJ04.aif	8118.344522	2	16236.689	63674	3.9216123
第9章	LoopChecker7.maxpat	tsugaru6.aif	11795.011121	1	11795.011121	46260	3.9219971
第9章	LoopChecker8.maxpat	momoe7.aif	3843.333263	2	7686.6664	30150	3.9223765
第9章	LoopChecker9.maxpat	momoe7.aif	3843.333263	4	15373.332	60290	3.9217262

Fig.1 metabar check table.

以上を考慮して、ここまでに登場した6種類の音素材から選んで、パラメータとして「3.92083」をカットアンドトライで決定して作ったのが、無限加速パッチ[39]であり、対応した無限減速パッチも制作した(第12章)。そして最後にパッチ[40]を作ってみたが、これはステレオ(定位は動かさずに固定)として、Lチャンネルが無限加速のRisset Rhythm、Rチャンネルが無限減速のRisset Rhythmとしているのだが、面白いことにウインドウ最下段のバークラフで「metabar中での進行具合」を表示してみると、その両端のあたりで左右の音像が急に中央に移動する。これこそ第5章の最後に確認したRisset Rhythmの本質そのものであり、**オクターブの螺旋が連結するために、途中ではステレオで分離しているのに、ちょうどmetabarの端っことでは両者がぴったり合っている、という事実をスッキリと体感できる。**このパッチ[40]を用いてここまでに登場した6種類の素材ループとともにそれぞれ約2分間のサウンドファイルとしたものが、サウンド[41-46]である。是非、ヘッドホンかイヤホンでこの不思議なサウンドを体験してみたい。

8. Real Time Risset Rhythm Generator

前章までで、いわば「Risset Rhythmのおさらい」は完了した。与えられた任意のループ素材(サウンドファイル)を使用して、それを使ったRisset Rhythmを生成することは、ほぼ定型作業として確立した。「面白いRisset Rhythmのサンプル」を作り出すことはこれで可能となったものの、ここで終わりでは面白くない。そこで第13章では、大きく目標を

「Real Time Risset Rhythm Generator」と掲げて、これに向けた実験の最初の一步を進めてみた。

ここでのパッチ[47-51]では、ループ素材としての波形バッファメモリをびったり「2000msec」(96000サンプル/48kHzサンプリング)と定めているために、metabarの誤差がまったく無いという環境とした。そして、Max ユーザにはお馴染みの「drumLoop.aif」という素材から、バスドラム、スネアドラム、裏打ちスネア、オープンハイハットなどの断片を「peek~」で切り出して、ビートに対応した波形バッファメモリのちょうどタイミングのところに貼り込むことで、刻々ときちんとしたループ素材のRisset Rhythmが生成される。さらにポイントとしては、4回=8秒間のループ素材から生成されるmetabar(31.36664秒)の区間の途中3箇所、「drumLoop.aif」からの断片を「peek~」で切り出して波形バッファメモリに貼り込むパターンを刻々と、randomにさじ加減を変えつつ変更している。randomで選ばれる候補の位置はいずれもビートに乗っているのでヘンになる事はないが、走らせてみると途中でパターンが変わることで、ある意味「飽きがこない」Risset Rhythmとなっている。ほんの一例であるが、モノラルで「永遠に加速する」というサンプル(約3分)がサウンド[52]であり、ステレオにして片方は永遠に加速しつつもう片方は永遠に減速しているサンプル(約3分)がサウンド[53]である。特に後者[53]において、metabarの周期ごとに、加速してきたパターンと減速してきたパターンが中央定位のあたりで一致する、という繰り返しがはっきりと聞こえて興味深いものになった。さじ加減を加えたrandomアルゴリズムに従ってリズムパターンは刻々と変化するので、このRisset Rhythmはまさに「ライブ生成されている」ことになる。

9. おわりに

温故知新として「Risset Rhythm」を追求してみることで、筆者の専門科目「音楽情報科学」を受講する学生とともに、あらためて筆者もこのRisset Rhythmをどっぷり体験することが出来た。音楽の要素のうち、メロディーとハーモニーについては、テンポが刻々と加速すればピッチが連続的に上昇し、テンポが刻々と減速すればピッチが連続的に下降するので、オクターブ違いのフレーズは2の累乗で繰り返し重複することから、局所的にはメロディーもハーモニーも壊れている筈だが、人間の聴覚的な知覚認知は音楽的な文脈を過去の反復や未来の推定によって補ってくれるので、音響は完全に破壊されることなく微妙に許容されている。実験の最後あたりに検討した「Real Time Risset Rhythm Generator」の考え方は、おそらくライブComputer Musicのパフォーマンスにおいて、人間の即興的な演奏フレーズをライブサンプリングしつつRisset Rhythm化することで、過去の音楽には

無かった「常にテンポが連続的に加速/減速する音楽」という新たな境地に発展する可能性を秘めている。筆者もいずれトライしてみたいと考えているが、興味ある方々の挑戦もおおいに歓迎/応援したい。

参考URL

1. https://youtu.be/oQf_tS5WAP4
2. <https://nagasm.org/1106/mac/RissetRhythm.mp3>
3. <https://nagasm.org/ASL/paper/SIGMUS202008.pdf>
4. <https://nagasm.org/ASL/RissetRhythm/fig/001.jpg>
5. <https://nagasm.org/ASL/RissetRhythm/fig/002.jpg>
6. <https://nagasm.org/ASL/RissetRhythm/sound/005.mp3>
7. <https://nagasm.org/ASL/RissetRhythm/sound/007.mp3>
8. <https://nagasm.org/ASL/RissetRhythm/fig/009.jpg>
9. <https://nagasm.org/ASL/RissetRhythm/fig/011.jpg>
10. <https://nagasm.org/ASL/RissetRhythm/sound/010.mp3>
11. <https://nagasm.org/ASL/RissetRhythm/sound/011.mp3>
12. <https://nagasm.org/ASL/RissetRhythm/sound/012.mp3>
13. <http://c4dm.eecs.qmul.ac.uk/papers/2011/Stowell2011icmc.pdf>
14. <https://nagasm.org/1106/EUtour2007/>
15. <https://nagasm.org/ASL/RissetRhythm/fig/015.jpg>
16. <https://nagasm.org/ASL/RissetRhythm/sound/014.mp3>
17. <https://nagasm.org/ASL/RissetRhythm/sound/015.mp3>
18. <https://nagasm.org/ASL/RissetRhythm/sound/016.mp3>
19. <https://nagasm.org/ASL/RissetRhythm/sound/017.mp3>
20. <https://nagasm.org/ASL/RissetRhythm/sound/018.mp3>
21. <https://nagasm.org/ASL/RissetRhythm/sound/019.mp3>
22. <https://nagasm.org/ASL/RissetRhythm/sound/020.mp3>
23. <https://nagasm.org/ASL/RissetRhythm/sound/021.mp3>
24. <https://nagasm.org/ASL/RissetRhythm/sound/022.mp3>
25. <https://nagasm.org/ASL/RissetRhythm/sound/023.mp3>
26. <https://nagasm.org/ASL/RissetRhythm/sound/>

- 024.mp3
27. <https://nagasm.org/ASL/RissetRhythm/sound/025.mp3>
28. <https://nagasm.org/ASL/RissetRhythm/sound/026.mp3>
29. <https://nagasm.org/ASL/RissetRhythm/sound/027.mp3>
30. <https://nagasm.org/ASL/RissetRhythm/fig/016.jpg>
31. https://nagasm.org/ASL/RissetRhythm/sound/up_down.mp3
32. https://nagasm.org/ASL/RissetRhythm/sound/down_up.mp3
33. <https://nagasm.org/ASL/RissetRhythm/fig/018.jpg>
34. <https://nagasm.org/ASL/RissetRhythm/sound/028.mp3>
35. <https://nagasm.org/ASL/RissetRhythm/fig/019.jpg>
36. <https://nagasm.org/ASL/RissetRhythm/sound/029.mp3>
37. <https://nagasm.org/ASL/RissetRhythm/fig/020.jpg>
38. <https://nagasm.org/ASL/RissetRhythm/sound/030.mp3>
39. <https://nagasm.org/ASL/RissetRhythm/fig/021.jpg>
40. <https://nagasm.org/ASL/RissetRhythm/fig/022.jpg>
41. <https://nagasm.org/ASL/RissetRhythm/sound/031.mp3>
42. <https://nagasm.org/ASL/RissetRhythm/sound/032.mp3>
43. <https://nagasm.org/ASL/RissetRhythm/sound/033.mp3>
44. <https://nagasm.org/ASL/RissetRhythm/sound/034.mp3>
45. <https://nagasm.org/ASL/RissetRhythm/sound/035.mp3>
46. <https://nagasm.org/ASL/RissetRhythm/sound/036.mp3>
47. <https://nagasm.org/ASL/RissetRhythm/fig/023.jpg>
48. <https://nagasm.org/ASL/RissetRhythm/fig/024.jpg>
49. <https://nagasm.org/ASL/RissetRhythm/fig/025.jpg>
50. <https://nagasm.org/ASL/RissetRhythm/fig/026.jpg>
51. <https://nagasm.org/ASL/RissetRhythm/fig/027.jpg>
52. <https://nagasm.org/ASL/RissetRhythm/sound/037.mp3>
53. <https://nagasm.org/ASL/RissetRhythm/sound/038.mp3>