

VisualStimuliProducer リリースノート

(視覚実験刺激簡易作成アプリケーション)

バージョン 0.9 - 2009/5/30

目次

1	概要	2
1.1	目的	2
1.2	環境	2
2	機能・操作	2
2.1	外観	2
2.2	コントロールパネルの操作	3
2.2.1	プリミティブの選択・配置	3
2.2.2	ユーザ定義の関数	4
2.2.3	刺激の調整	5
2.3	刺激呈示領域の操作	5
3	プリミティブの詳細	6
3.1	共通パラメータ	6
3.2	circle	6
3.3	rectangle	6
3.4	grating	7
3.5	circular grating	7
3.6	image	8
3.7	gabor	8
3.8	gauss	9
3.9	gauss filter	9
4	使用例	10
4.1	重ね合わせ	10
4.2	ランダムドット	10
4.3	フラッシュ、フリッカなど	11
5	既知の問題など	11

1 概要

1.1 目的

VisualStimuliProducer は、視覚科学の研究でよく用いられる実験刺激を簡単に作成・評価することができるアプリケーションです。自分で現象をすぐに確認したいときや、思いついたアイデアをその場で試したいときなどに使用されることを想定しています。したがって、本格的な実験を行うことには適していません。

1.2 環境

ActionScript3.0 で実装しています。実行には Adobe Flash Player が必要です。実行方法には二通りの方法があります。一つ目は、web ブラウザのプラグインを利用してブラウザ上で実行する方法です。現在 (2009/5/30)、<http://objective.oteage.net/VisualStimuliProducer/> にアクセスすることで利用できます。二つ目の方法は、スタンドアロンの Flash Player を用いてローカルで実行する方法です。Flash Player は <http://www.adobe.com/support/flashplayer/downloads.html> からダウンロードすることができます。使用している OS に合った Flash player をダウンロードした後、<http://objective.oteage.net/VisualStimuliProducer/VisualStimuliProducer.swf> をダウンロードしてください。この swf ファイルを Flash Player で開くことで実行することができます。

2 機能・操作

2.1 外観

図1のような GUI で操作を行います。画面左がコントロールパネル、右が刺激の配置・呈示領域です。

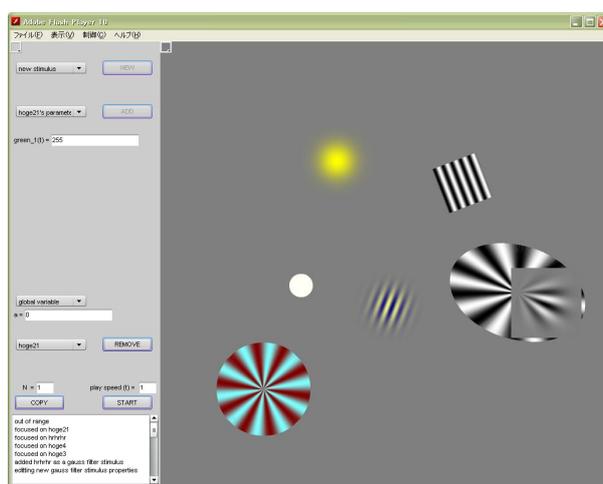


図 1: アプリケーションの外観

2.2 コントロールパネルの操作

2.2.1 プリミティブの選択・配置

左上に "new stimulus" と書かれたコンボボックスがあります。クリックすると、使用できる刺激プリミティブのリストが図 2 のように表示されます。

この中から一つ選ぶと、図 3 のように刺激の名前の入力待ち状態になります。テキストボックスに適当な名前を入力して "NEW" ボタンを押すと、刺激が配置されます。次に、刺激のパラメータの設定を行います。



図 2: 刺激プリミティブ

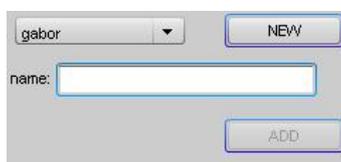


図 3: 名前の入力

"~ 's parameter" をクリックすると、配置した刺激の属性リストが表示されます (図 4)。それぞれ、後に説明するユーザ定義の関数および時間の関数として、MATLAB ライクな表記で刺激の属性を自由に設定することができます。"ADD" ボタンのクリックで設定が完了します。刺激の属性は後にいつでも変更することができます。



図 4: 刺激パラメータの設定

2.2.2 ユーザ定義の関数

"global variable" をクリックすると、ユーザが自由に設定できる変数リスト (アルファベット $a \sim z$ から e, h, t, w を除く 22 種類) が現れます (図 5)。初期値は全て 0 です。これらには次の規則で数式を適用することができます。

演算子

二項演算子は次の 6 種類が用意されています。

+ : 加算、- : 減算、* : 乗算、/ : 除算、% : 剰余、^ : 累乗。

数学関数

あらかじめ定義されている関数は次のとおりです。すべて引数は 1 つです。関数名の入力は半角小文字に限ります。

- floor() : 小数点以下の切り下げ。
- ceil() : 小数点以下の切り上げ。
- rand() : 0 から引数までの任意の実数。
- sin(), cos(), tan() : 正弦、余弦、正接。引数の単位は degree。
- asin(), acos(), atan() : 逆正弦、逆余弦、逆正接。返り値の単位は degree。
- abs() : 絶対値。
- exp() : 自然対数の底を底とする指数関数。
- ln() : 自然対数。
- sqrt() : 平方根。
- round() : 小数点以下の四捨五入。
- irand() : ムービー開始時に一度だけ実行される rand()。

定数

4 種類の変数が用意されています。入力は全て半角小文字に限ります。

- e : 自然対数の底 (約 2.718282)。
- pi : 円周率 (約 3.141592)。
- w : 刺激呈示領域の幅 [pixel](768)。
- h : 刺激呈示領域の高さ [pixel](768)。

単に数値を入力すればそれも定数として扱われます。

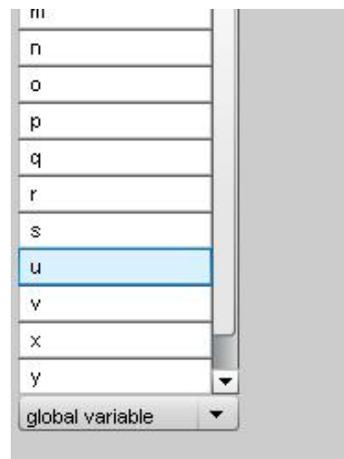


図 5: global variable

変数

時間を表す値には t を用います。単位は秒です。また、別に設定した変数 (アルファベット 1 文字) も数式に含めることができます。ただし自分自身を含めることはできません。変数を階層的に設定することで、変数の見通しが良くなります。

刺激のパラメータに記述する数式の形式もこれと同様です。入力はすべて半角小文字に限ります。刺激パラメータの設定には、別に設定した変数を用いても、変数を用いずに直接に式を書いてもどちらでも構いません。設定した変数、刺激パラメータはムービー実行中に繰り返し評価され、描画に反映されます。

2.2.3 刺激の調整

刺激はいくつでも追加することができます。配置した刺激のリストはパネル左下の "stimulus list" から見ることができます。ここから再設定したい刺激を選択すると、パラメータを設定するコントロールが再び表示されます。刺激を削除したいときは、刺激がフォーカスされた状態で "REMOVE" ボタンを押します。刺激のコピーを行いたいときは、左下の " $N =$ " にコピーの個数を入力して "COPY" ボタンを押します。ムービーを実行するときは、実行速度倍率 (右下の "play speed = ") を確認してから "START" ボタンを押します。ムービーの実行中であっても、刺激の設定をダイナミックに変更することができます。変更はすぐに描画に反映されます。



図 6: 刺激のコントロール

なお、設定にミスなどがあった場合はプロンプトにその都度メッセージが表示されますので、デバッグを行ってから再度 "START" を押してください。

2.3 刺激呈示領域の操作

配置した刺激はマウスでも操作することができます。マウスドラッグによって刺激の位置を変更できます。また、刺激をポイントした状態でのマウスホイール操作で刺激の回転を行うことができます。また、shift キーを押した状態でホイールを操作すると縦方向の拡大縮小、ctrl キーを押した状態でホイールを操作すると横方向の拡大縮小を行うことができます。背景色は、画面左上の小さな正方形で変更できます。クリックすると色を自由に選択できるようになります。なお、コントロールパネルの色も同様にカスタマイズできます。

3 プリミティブの詳細

8種類の基本的な刺激が用意されています。それぞれのパラメータを、初期値と範囲と合わせて順に紹介します。なお、空間の単位は [pixel]、時間の単位は [秒] となっています。

3.1 共通パラメータ

全種類の刺激に共通して含まれるパラメータです。

- $x(t)$: 横軸座標 (初期値 : 384、左端 : 0 → 右端 : 768)
- $y(t)$: 縦軸座標 (初期値 : 384、上端 : 0 → 下端 : 768)
- $alpha(t)$: 透明度 (初期値 : 1.0、透明 : 0.0 → : 不透明 1.0)
- $angle(t)$: 角度 (初期値 : 0、0 ~ 360)
- $scaleX(t)$: 横拡大率 (初期値 : 1.0、0 ~ 100)
- $scaleY(t)$: 縦拡大率 (初期値 : 1.0、0 ~ 100)

3.2 circle

円 (楕円) です。

- $Rx(t)$: 横半径 (初期値 : 20、0 ~ 400)
- $Ry(t)$: 縦半径 (初期値 : 20、0 ~ 400)
- $red(t)$: 赤階調値 (初期値 : 255、0.0 ~ 255)
- $green(t)$: 緑階調値 (初期値 : 255、0.0 ~ 255)
- $blue(t)$: 青階調値 (初期値 : 255、0.0 ~ 255)

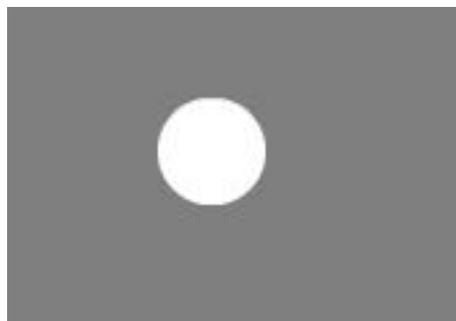


図 7: circle

3.3 rectangle

長方形です。

- $width(t)$: 横幅 (初期値 : 60、0 ~ 400)
- $height(t)$: 高さ (初期値 : 60、0 ~ 400)
- $red(t)$: 赤階調値 (初期値 : 255、0.0 ~ 255)
- $green(t)$: 緑階調値 (初期値 : 255、0.0 ~ 255)
- $blue(t)$: 青階調値 (初期値 : 255、0.0 ~ 255)

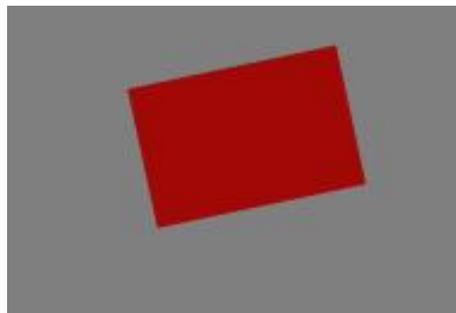


図 8: rectangle

3.4 grating

指定した二種類の色の間で正弦波上に変調する縞です。

- $width(t)$: 横幅 (初期値 : 80、0 ~ 1000)
- $height(t)$: 高さ (初期値 : 80、0 ~ 1000)
- $red_1(t)$: 赤階調値 1 (初期値 : 255、0.0 ~ 255)
- $green_1(t)$: 緑階調値 1 (初期値 : 255、0.0 ~ 255)
- $blue_1(t)$: 青階調値 1 (初期値 : 255、0.0 ~ 255)
- $red_2(t)$: 赤階調値 2 (初期値 : 0、0.0 ~ 255)
- $green_2(t)$: 緑階調値 2 (初期値 : 0、0.0 ~ 255)
- $blue_2(t)$: 青階調値 2 (初期値 : 0、0.0 ~ 255)
- $amp(t)$: 階調振幅係数 (初期値 : 1.0、0.0 ~ 1.0)
- $f_s(t)$: 空間周波数 (初期値 : 0.4、0.0 ~ 100)
- $f_t(t)$: 時間周波数 (初期値 : 10.0、-1000 ~ 1000)

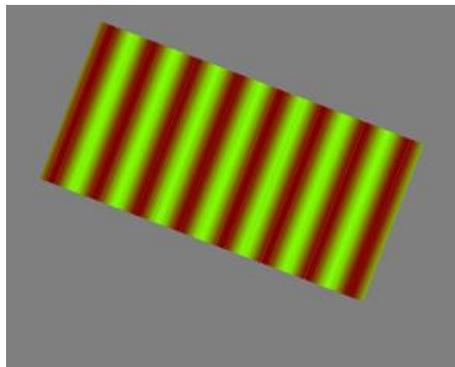


図 9: grating

3.5 circular grating

円形の grating です。

- $r(t)$: 半径 (初期値 : 80、0 ~ 1000)
- $red_1(t)$: 赤階調値 1 (初期値 : 255、0.0 ~ 255)
- $green_1(t)$: 緑階調値 1 (初期値 : 255、0.0 ~ 255)
- $blue_1(t)$: 青階調値 1 (初期値 : 255、0.0 ~ 255)
- $red_2(t)$: 赤階調値 2 (初期値 : 0、0.0 ~ 255)
- $green_2(t)$: 緑階調値 2 (初期値 : 0、0.0 ~ 255)
- $blue_2(t)$: 青階調値 2 (初期値 : 0、0.0 ~ 255)
- $amp(t)$: 階調振幅係数 (初期値 : 1.0、0.0 ~ 1.0)
- $f_s(t)$: 空間周波数 (初期値 : 10.0、0.0 ~ 100)
- $f_t(t)$: 時間周波数 (初期値 : 10.0、-1000 ~ 1000)

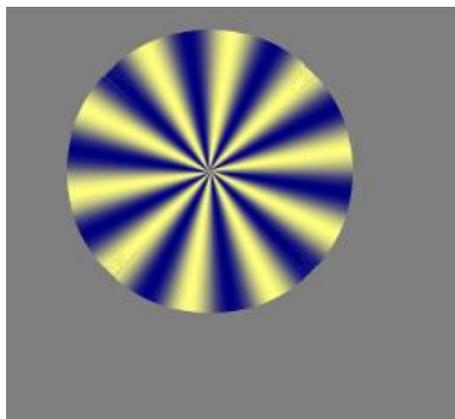


図 10: circular grating

3.6 image

画像 (JPG, GIF, PNG) を読み込んで使用できます。名前を入力するテキストフィールドに画像ファイルの URL を指定して"NEW"を押すと画像がロードされます。本アプリケーションをローカルで実行している場合は、swf ファイルからの相対パスまたは絶対パスで画像ファイルを指定することもできます。



図 11: image

3.7 gabor

ガボールパッチです。色の変調は *grating* と同様です。

- $width(t)$: 横幅 (初期値 : 160、0 ~ 1000)
- $height(t)$: 高さ (初期値 : 160、0 ~ 1000)
- $red_1(t)$: 赤階調値 1 (初期値 : 255、0.0 ~ 255)
- $green_1(t)$: 緑階調値 1 (初期値 : 255、0.0 ~ 255)
- $blue_1(t)$: 青階調値 1 (初期値 : 255、0.0 ~ 255)
- $red_2(t)$: 赤階調値 2 (初期値 : 0、0.0 ~ 255)
- $green_2(t)$: 緑階調値 2 (初期値 : 0、0.0 ~ 255)
- $blue_2(t)$: 青階調値 2 (初期値 : 0、0.0 ~ 255)
- $grating\ amp(t)$: 階調振幅係数 (初期値 : 1.0、0.0 ~ 1.0)
- $f_s(t)$: 空間周波数 (初期値 : 0.4、0.0 ~ 100)
- $f_t(t)$: 時間周波数 (初期値 : 10.0、-1000 ~ 1000)
- $gauss\ amp(t)$: ガウス強度 (初期値 : 1.0、0.0 ~ 1.0)

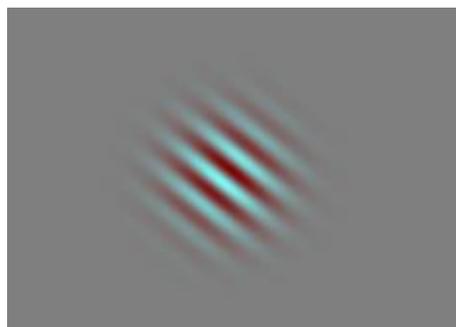


図 12: gabor

3.8 gauss

ガウスパッチです。指定した2つの色の間でガウス変調しています。

- $\sigma_X(t)$: 横軸 σ (初期値 : 20、0 ~ 1000)
- $\sigma_Y(t)$: 縦軸 σ (初期値 : 20、0 ~ 1000)
- $red_1(t)$: 赤階調値 1 (初期値 : 127、0.0 ~ 255)
- $green_1(t)$: 緑階調値 1 (初期値 : 127、0.0 ~ 255)
- $blue_1(t)$: 青階調値 1 (初期値 : 127、0.0 ~ 255)
- $red_2(t)$: 赤階調値 2 (初期値 : 255、0.0 ~ 255)
- $green_2(t)$: 緑階調値 2 (初期値 : 255、0.0 ~ 255)
- $blue_2(t)$: 青階調値 2 (初期値 : 255、0.0 ~ 255)
- $amp(t)$: ガウス強度 (初期値 : 1.0、0.0 ~ 1.0)



図 13: gauss

3.9 gauss filter

ガウスフィルタです。指定した色からの透明度をガウス変調しています。

- $\sigma_X(t)$: 横軸 σ (初期値 : 20、0 ~ 1000)
- $\sigma_Y(t)$: 縦軸 σ (初期値 : 20、0 ~ 1000)
- $red(t)$: 赤階調値 (初期値 : 127、0.0 ~ 255)
- $green(t)$: 緑階調値 (初期値 : 127、0.0 ~ 255)
- $blue(t)$: 青階調値 (初期値 : 127、0.0 ~ 255)
- $amp(t)$: ガウス強度 (透明度) (初期値 : 1.0、0.0 ~ 1.0)



図 14: gauss filter

4 使用例

4.1 重ね合わせ

透明度を操作することで、低周波と高周波の縞を重ねて表示することができます(図 15)。もちろんドリフトさせることもできます。

4.2 ランダムドット

irand() 関数とコピー機能を使うことでランダムドットを生成できます。運動透明視などの実験がすぐにできます(図 16)。

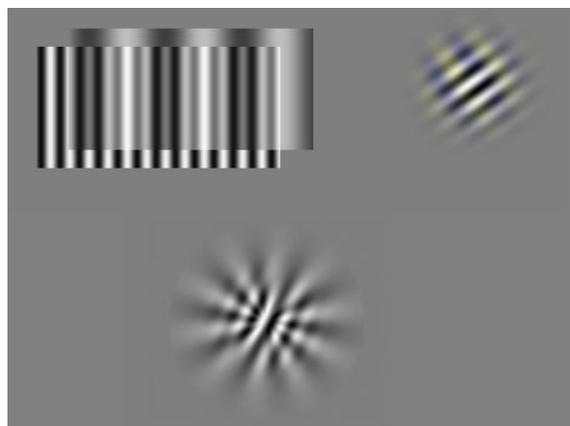


図 15: 重ね合わせ

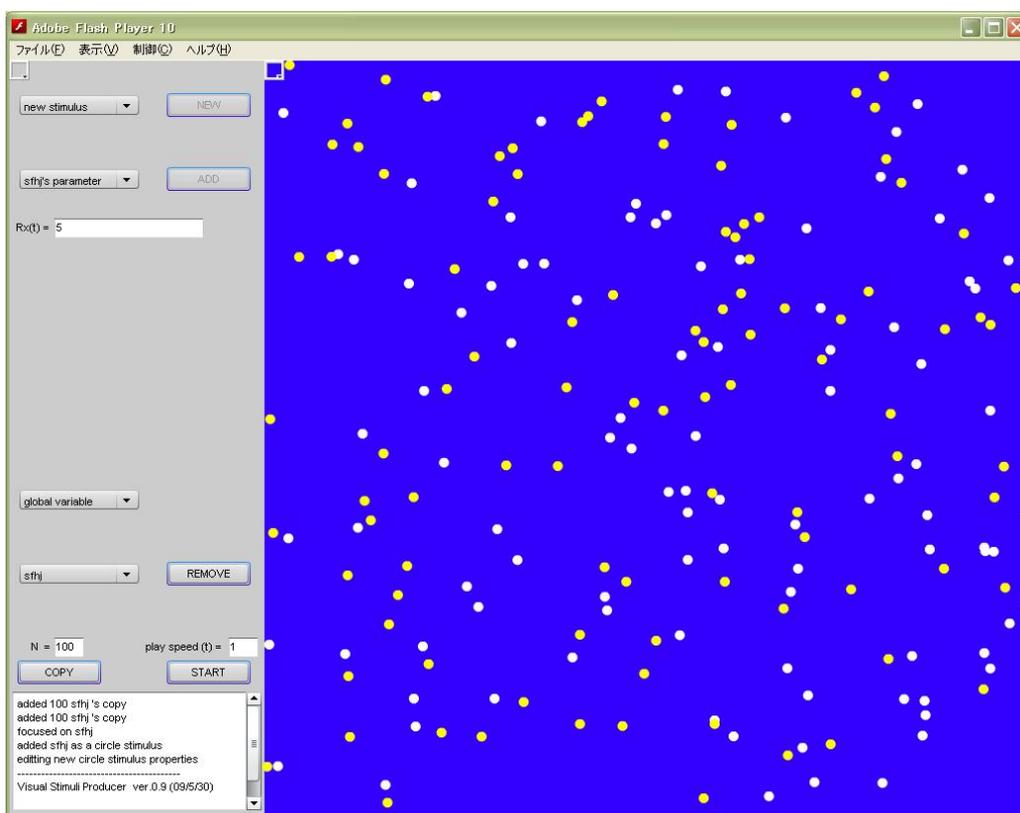


図 16: ランダムドット

4.3 フラッシュ、フリッカなど

色や alpha 値に $\sin()$ 関数を適用して時間変化させればフリッカ刺激を作成できます。 $\text{round}()$ 関数と組み合わせてフラッシュ刺激を作成することも簡単にできます。

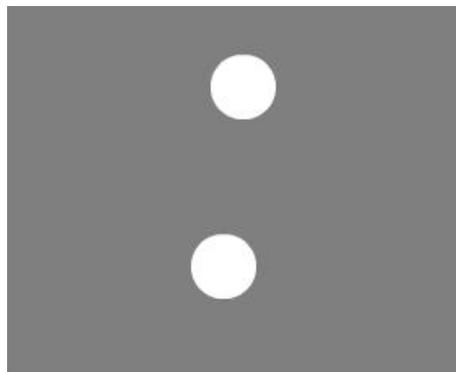


図 17: フラッシュラグの実験など

5 既知の問題など

本アプリケーションはフレームの同期制御を行っていません。また、gabor など刺激によっては計算量が多く動きが滑らかでない場合があります。加えて、現状では三次元の動きを表現することが非常に困難です。今後はこれらの問題を解決し、さらに状態の save & load を行う機能を追加する予定です。

ソースコードは <http://objective.oiteage.net/VisualStimuliProducer/src.zip> にて公開しています。