

Psychlops: C++ 言語による汎用的な視覚刺激提示ライブラリ

細川 研知*・丸谷 和史**・佐藤 隆夫***

* 東京大学インテリジェントモデリングラボラトリー

〒113-0033 東京都文京区弥生 2-11-16

** NTT コミュニケーション科学基礎研究所

*** 東京大学大学院人文社会系研究科

1. はじめに

視覚の心理物理学・視覚科学研究を志す学生・大学院生たちにとって、プログラミング技術の習得は最初の大きな壁となる。心理物理学実験を始めようとする、心理物理学そのものと同時に一般的なプログラムのスキル、さらに心理物理学実験に特化したソフトウェアに対応するスキルを習得する必要がある。さらに、これらの特殊なソフトウェアは外国で開発されたものが多く、国内でのサポート状況はよいとはいえない。また英語で書かれた辞書的なマニュアルは初期の習得時にはあまり役に立たない。

教育を受ける側だけではなく、教育を与える側の負担も大きい。我が国では心理学は文系の学部内に位置づけられており、そこへ入学する学生たちのコンピュータスキルは必ずしも高くない。卒業研究や修士論文のための研究を行う学生たちが心理学実験のために必要な特殊なツールを一通り使えるようになるには、大学における一般的な情報処理の授業だけでは不十分である。また、上で述べたような状況ではそれらの学生たちが独習を行うには学習の負担が大きく、研究室の側で人的な面を含んだ総合的な教育環境を準備し、機材の面では近年頻繁に行われる機材の世代交代に対応してこれを維持する必要がある。この教育環境の整備・維持にかかる大きな金銭的・人的なコストが研究全体の質に与える影響は小さくないと思われる。

初学者がより簡単に実験刺激の作成を行える

ように、筆者らは新しい心理学実験ツール“Psychlops”の開発を行っている。Psychlopsの基本的な性能・精度は、現在広く使われているツールと比較して遜色ない。本稿ではPsychlopsの基本性能とその周辺環境について、他の一般的なツールとの比較をしながら紹介していく。

2. 初学者のための環境整備

上に述べたように、初学者の学習を考えたときにはツールそのものが本格的な心理物理学実験に堪える精度を持つだけでなく、このツールをコアとした総合的な環境が提供される必要がある。Psychlopsにおけるこうした環境

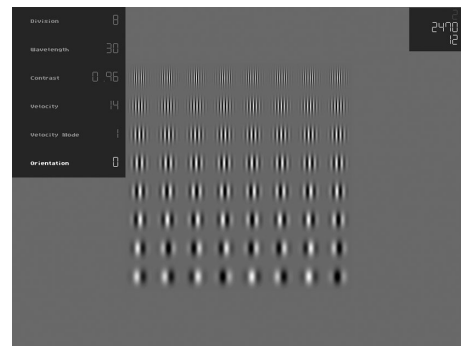


図1 各種支援ツールの実行例。左上 プログラム中の任意の変数を実行中に変更するインターフェース。変更できる変数の名称と現在の値が表示されている。このインターフェースはキーボードから操作することができ、変数1つにつき1行追加するだけで作成可能である。右上 コマ落ち監視ツールの実行例。上段はコマ落ちしたフレーム数、中段は総表示フレーム数、下段はフレーム間隔をミリ秒単位で示している。

の整備はまだ始まったばかりであり、これから充実させるべき部分も多い。しかし、現状でも既存のツールにはない特色を持っている。

2.1 日本語マニュアル・チュートリアル of 充実

Psychlops のマニュアルは初学者のスキル習得の助けとなることを意識して作られている。単なる関数リファレンスに加えて、基本的な描画から実際の心理物理学実験の研究計画の実装までのチュートリアルが含まれる。C++ による開発環境を整えるのはプログラミング初心者にとっては難しいが、このマニュアルではその点についても図表を含めて丁寧に解説されており、初学者であっても1時間前後で Psychlops による開発環境を整えることができる。また、Q&A や関数逆引き、Tips の項目が数多く用意されており、ユーザの様々な要望に対応している。このマニュアルは、後述の専用ウェブサイトから入手できる。また、専用ウェブサイト “Psychlops Wiki” (<http://psychlops.l.u-tokyo.ac.jp/>) ではライブラリ本体の配布のほか、支援ツールや技術情報の提供、フィードバックの受付を行っている。これらのフィードバックは随時マニュアルに追加される予定である。

2.2 デモンストレーション環境の整備

心理物理学実験の計画段階では、対象となる現象について、刺激のパラメータをさまざまに変えて観察することで現象の理解が進みよりよい実験計画を立てることが可能になる。そこでプログラム中の変数を実行時に操作するための簡易 GUI 環境を作成した。この環境は実験用プログラムに数行追加するだけで容易に利用できる (図 1)。また、初学者がプログラムを書く際には、すべて一から書き始めることは難易度が高く、手本をもとに徐々に馴染むほうが容易であると考えられる。そのために、典型的な視覚刺激のコードと実行ファイルが数十個準備され、視覚科学データベース “Visiome Platform”¹⁾ や “Psychlops Wiki” などのインターネット上のサイトから入手できる (図 2)。初学者はこれらのサイトから自分が作りたいものの例を検索し、題材となっている現象を理解するとともにプロ

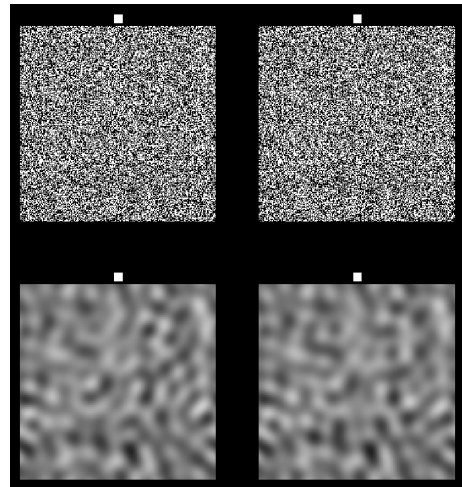


図 2 Psychlops による刺激例。ライダムノイズによるステレオグラムと、それをローパスフィルタリングしたものを示す。ユーザコードはサンプルコードを利用して約 50 行に収まっている。

グラムの方法についても自学自習することができる。

この機能は学会発表等における刺激、特に動画的な刺激のデモ作成にも有用である。学会などで実験に用いたものと近い刺激をデモンストレーションできる意義は大きい。また近年は、アウトリーチ活動の奨励など、実験刺激のデモンストレーションの配布の必要性が高まっている。Psychlops は一つのソースコードから Windows と MacOSX 環境で使えるデモンストレーションの実行ファイルを作成でき、デモの作成環境としては他のツールに対して優れたものとなっている。

3. Psychlops の基本性能

初学者によって学習されたスキルが無駄にならないためには、ライブラリが本格的な心理物理学研究に用いることが可能であり、かつ様々な機器環境で継続的に使用できることが望ましい。Psychlops は前節で述べたような特色だけではなく、その基本性能についても他のツールと比較しうる水準に達している。

3.1 ソースコードの互換性と簡略化

Psychlops は現在最も一般的な開発言語の一つである C++ 言語のライブラリである。C++ という習得が難しいという印象があるが、高度の機能を使いこなす必要がなければ習得は難しくない。また、大学教養課程におけるプログラミングの授業で C++ の基本となる C 言語が多く用いられており、学生たちが学部進学時に最も初歩的な部分や基本的な数値計算の実装方法の知識を持っていることが期待できる。さらに、C++ 言語は汎用性とプログラムの実行速度の点において他のスクリプト系のプログラミング言語に比べ優れている。

しかし、Windows や Macintosh などの現在一般的に手に入るシステム上で心理学実験を行うにはオペレーティングシステム (OS) に依存した特殊なプログラミングを行う必要があり、特に初学者には負担が大きい (図 3)。

Psychlops は汎用的な描画ライブラリの一つである OpenGL を基本に作成されているが、簡略化と互換性の拡張を行うために、描画に必要な OS への指令を全て隠蔽している。たとえば、Windows, Macintosh どちらでも Canvas クラスと呼ばれる抽象化された変数を宣言するだけで描画のためのウィンドウが確保される (図 3)。この簡略化と並行して、OpenGL が持つ機種互換性がさらに拡張されている。現段階において Psychlops では Windows, Macintosh 向けに描画部分だけではなくキーボードやマウスの入力取得、データのファイル出力も含むソースコードの完全互換性が実現されている。

3.2 Psychlops における基本精度の実現

Psychlops は、特殊な機器なしで汎用のコンピュータと OS での正確な描画を可能としている。心理物理学実験を行うためのツールとしては、(1) 正確な大きさで描画すること、(2) 正

(a) OpenGL + Windows WGL	(b) GLUT	(c) Psychlops
<pre>int main(int argc, char *argv[]) { WNDCLASSEX wcx; HDC the_display_; HWND the_window_; HGLRC the_GL_context_; DEVMODE mode_, original_mode_; int width_, height_, colordepth_; double refreshrate_; DEVMODE &d_mode; memset(&mode_, 0, sizeof(mode_)); memset(&original_mode_, 0, sizeof(original_mode_)); mode_.dmSize = sizeof(mode_); mode_.dmPelsWidth = d_width; mode_.dmPelsHeight = d_height; mode_.dmBitsPerPel = d_colordepth; mode_.dmDisplayFrequency = d_refreshrate; mode_.dmFields = DM_BITSPERPEL DM_PELSWIDTH DM_PELSHEIGHT DM_DISPLAYFREQUENCY; wcx.cbSize = sizeof(WNDCLASSEX); wcx.style = CS_GLOBALCLASS; wcx.lpfnWndProc = (WNDPROC) &proc; wcx.cbClsExtra = 0; wcx.cbWndExtra = 0; wcx.hInstance = startupinfo.hInstance_; wcx.hIcon = NULL; wcx.hIconSm = NULL; wcx.hCursor = NULL; wcx.hbrBackground = NULL; wcx.lpszMenuName = "Psychlops Window"; wcx.lpszClassName = startupinfo.pClassName; if (!RegisterClassEx(&wcx)) Exception("Failed."); if (ChangedDisplaySettings(&d_mode, CDS_FULLSCREEN) != DISP_CHANGE_SUCCESSFUL) throw Exception("Error"); width_ = d_mode.dmPelsWidth; height_ = d_mode.dmPelsHeight; (以下略) }</pre>	<pre>void display(void) { glClear(GL_COLOR_BUFFER_BIT); glRasterPos2d(100,100); const char HELLOWORLD[14] = "Hello, world!"; for(int i=0; i<14; i++) glutBitmapCharacter (GLUT_BITMAP_HELVETICA_10, HELLOWORLD[i]); glFlush(); } void keyboard(unsigned char key, int x, int y) { switch (key) { case '\033': glutLeaveGameMode(); exit(0); break; default: break; } } int main(int argc, char *argv[]) { glutInit(&argc, argv); glutInitDisplayMode(GLUT_RGBA GLUT_DOUBLE); glutGameModeString ("width=1024 height=768 bpp=24 hertz=60"); glutEnterGameMode(); glutDisplayFunc(display); glutKeyboardFunc(keyboard); glutViewport(0, 0, 1024, 768); glOrtho(0, 1024, 0, 768, -1000, 1000); glutMainLoop(); return 0; }</pre>	<pre>void psychlops_main() { Canvas display(1024, 768, 32, 60.0); display.msg("Hello, world!", 100, 100); display.flip(); while(!Input::get(Keyboard::esc)) {} }</pre>

図 3 OpenGL+Windows WGL (左), GLUT (中), Psychlops (右) での最小の “Hello World” コード例。Windows WGL では 200 行近く必要である。GLUT は空白、括弧を除いて 26 行で構成されており、フルスクリーンイベント駆動型プログラムとして実行される。Psychlops では Canvas クラスへの機能集約により空白、括弧を除いて 5 行で構成されており、GLUT に比べ大幅に少ない。またイベント駆動型ではなく手続き的記述通りに実行される。

確な輝度で描画すること、(3) 正確なタイミングで描画すること、の3つが必要条件である。Psychlopsでは以下に述べるような基本精度の実現が行われている。

3.2.1 描画サイズの精度保証

現代のコンピュータは画素を視覚的情報提示の表示単位としており、刺激を提示する大きさは表示する画素の物理的大きさで決まる。PsychlopsはOpenGLが定める方式²⁾に従って画素を単位とした正確な描画サイズを保証している。ユーザは画素の物理的大きさを基準として正確な大きさでの描画を行うことができる。

3.2.2 輝度の精度保証

PsychlopsではCRTとプログラム上の輝度指定値を線形に保つためGamma補正と呼ばれる手続き³⁻⁷⁾を自動的に適用する命令が用意されており、予め測定したGamma値を指定することで簡単に補正することができる(図4)。またGamma値を測定する簡単なツールも用意している。

Psychlopsでの輝度精度は、使用するハードウェアの輝度精度に拘束される。市販されている汎用的なビデオボードでは、RGB各原色の輝度は8ビットの精度であることが多いので、特

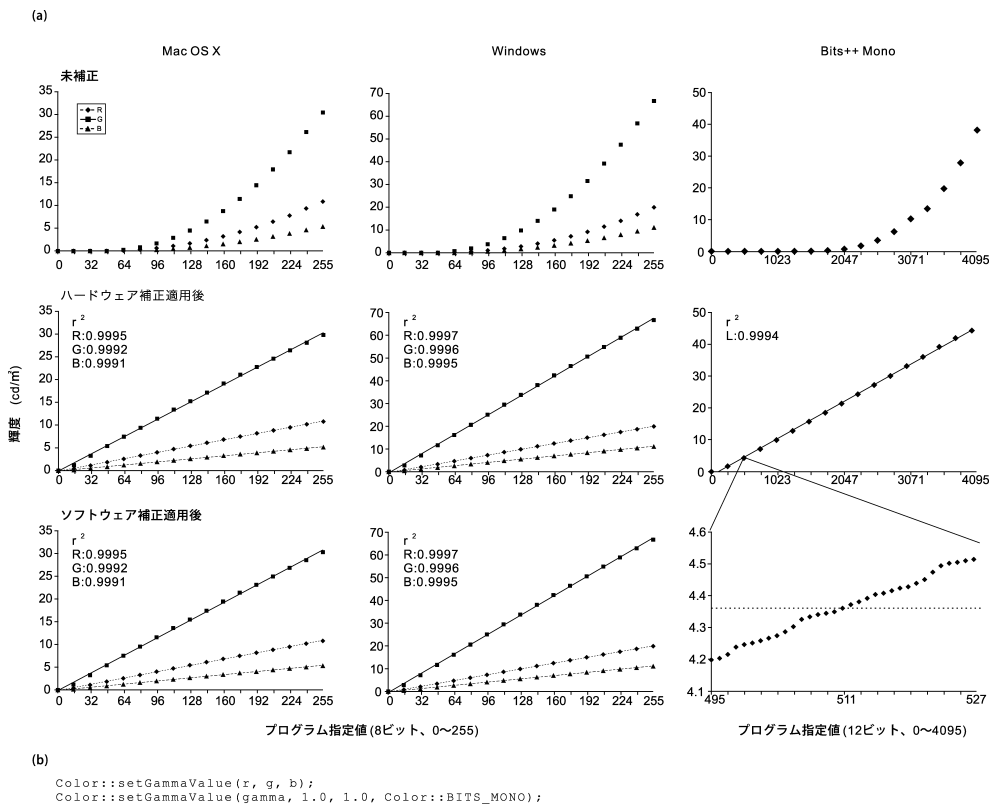


図4 Psychlopsにおけるガンマ補正。(a) 横軸を理論値、縦軸を輝度 (cd/m^2) にとり、両者の関係を示す。(上) 校正前の理論値-輝度値関係には、非線形な歪みがある。(中) ハードウェアガンマ校正を適切に適用した結果、理論値-輝度値関係は線形に修正されている。(下) ハードウェアによる校正ができない場合、自動的にソフトウェアエミュレーションによって校正に切り替わる。この場合でもハードウェア校正と同様に線形に修正されている。(右下) Bits++Monoモードでの、最大輝度の25%付近での詳細な理論値-輝度関係。横軸の両端間は8ビット精度における2段階分に相当する。Bits++は本来14bitの精度があるが、測定器における誤差を考慮して12ビット精度で測定した。Bits++の使用により少なくとも12bit精度が達成されていることが分かる。(b) ガンマ補正命令のコード例。r, g, bには浮動小数点でディスプレイのガンマ特性を入力する。Bits++Monoモードにするには、第一引数にガンマ値を、第四引数にモード変更指定をするだけで他の部分を変更せずにBitsモードとして動作する。

殊なカードを利用せず Psychlops を用いた場合のシステム全体としての輝度精度は 8 ビットとなる。Psychlops では Cambridge Research Systems 社製 Bits++ を用いた 14 bit 表示を可能にするための命令オプションも実装されている。図 4 に示したようなオプションをガンマ補正時に指定すると、Psychlops 上での輝度指定値が Bits++ によって 14 bit 精度で表示される。コントラスト感度測定などの精度の高い描画が必要な状況でもプログラムを改変する必要がなく、実験環境が共有の場合でも基本となるプログラムを個人の汎用的な環境で作成することができる。

ビデオボードで RGB の各値を 8 ビットに丸め込む際、ハードウェアごとに丸めこみの方法が異なるため、表示結果にばらつきが生じることがある。Psychlops ではこれを強制的に IEEE 754 に従った偶数丸めを行うことで正確な演算を保証している。

3.2.3 時間精度の保証

コンピュータ用表示装置上で実験刺激を提示する際には、リフレッシュフレーム単位での制御を行うことになる。この制御には、1 フレーム時間内にディスプレイの一部分だけが描き換わってしまわないようにすること（表示装置の垂直同期信号に同期して描き換えが行われること）と、表示内容の描き換え、バッファ切り替えが正しい（ユーザの意図する）フレーム数単位の時間間隔で行われることが必要である。

2000 年代以降のビデオボードでは、ダブルバッファリングと共に、バッファ（フレーム）切り替えを垂直同期信号に対して同期させる機能を備えている。Psychlops はこの機能を活用しており、プログラム側ではフレーム切り替えを命令のタイミングを気にする必要はなく、自動的に垂直同期信号と同期してフレームの切り替えが行われる（図 5）。したがって、正しく設定された環境で Psychlops を用いれば、意図したとおりの時間精度で表示される。

しかし、こうした方法でも、実際のバッファ切り替えのタイミングを完全に保証することは

できない。例えば、複雑な刺激の描画では計算量が増え、所期のフレーム切り替えのタイミングまでに描画が終了しない事態が起こりうる。その場合、命令をビデオボードに送るタイミングにも遅れが生じ、新しい刺激の提示のタイミングにも遅れが生じる。描画計算に必要な時間はコンピュータの処理速度に依存するが、ひとつのアプリケーションが中央演算装置 (CPU) を独占できないプリエンプティブマルチタスク方式と呼ばれる近代的な OS では、煩雑な手続きなしでコンピュータの処理速度を完全に予測することはできない^{8,9)}。従って、このような計算量の増加に伴うバッファ切り替えの遅延（コマ落ち）を完全に防ぐことは難しい。しかしユーザの意図したフレーム間隔と実際のプログラムの挙動の食い違いを監視し、必要に応じてユーザ側に警告することは可能であり、Psychlops

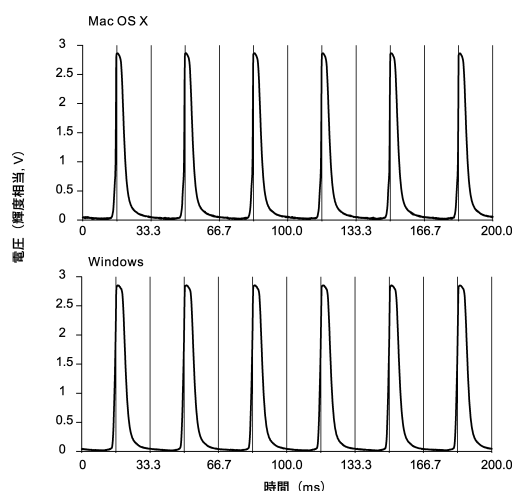


図 5 Psychlops を用いた刺激提示の時間精度の測定結果。30 Hz のフリッカを表示するプログラムを実行した際の CRT の輝度をフォトトランジスタを用いて計測した実測値の一部を示す。横軸は時間、縦軸は抵抗器両端の電圧である。横軸補助線は垂直同期信号の間隔を示す。蛍光体が陰極線に走査されてからの輝度変化の特性により櫛状の波形が表れており、陰極線の走査間隔が一定であることが分かる。輝度のピーク値の間隔を 1 分間 (1800 回) 計測したところ、輝度の変化周期は 1/30 秒 \pm 1/6000 秒 (垂直同期周期の 1%) 以内の範囲にすべておさまっており、正確な周期で実行されていた。

はこの機能を実装している。この機能は通常のプログラムに所定の命令を2行分追加するだけで利用することができ、これによりユーザはプログラム実行時の各瞬間に対して適切な計算量を知ることができる。

3.3 ソースコードの柔軟性と広いユーザーレベルへの対応

初学者向けのツールでは、なるべく簡単に習得できるよう、簡略化された命令セットを用いることが多いが、複雑な刺激を作成するには相応の命令セットが必要である。Psychlopsではプログラミングの経験が浅い初学者から、高度なプログラミング技術を持った習熟者まで幅広く対応できるよう努めた。たとえば、パラメタを細かく指定できる命令であっても、常用される値がある場合はそれをデフォルト値として詳細な指定を省略可能とし、特殊な場合にパラメタを追加指定する方式を採用した。このように、命令のレベルを多層的に構成することによって、ユーザの目的に対する細かな習熟度の違いまで含めた対応を実現した。

また、C/C++がもつ豊富な資産を組み合わせにより柔軟な刺激生成を行うこともできるように留意した。たとえば、高速フーリエ変換用のライブラリとして広範に利用されているFFTW¹⁰⁾とPsychlopsを結合して利用することができる。これらのプログラム例は、Psychlops配布サイトおよびマニュアルで公開している。また、多くの特殊機器のドライバライブラリ類はC言語で提供されており、こうした特殊機器用のライブラリをPsychlopsと併用して刺激提示と同期して制御できる。脳波や生理学実験では、刺激が実際に提示されたタイミングを何らかのトリガとしてシステムの外に出力する必要があることが多い。このようなトリガを出力する例として、National Instruments社のA/DコンバータをPsychlopsから使用するプログラムの例をマニュアルで紹介している。さらに、OS固有の命令群もC言語で利用可能なため、たとえば画面提示の時間制御だけをPsychlopsによって行い、描画命令にOpenGL命令のみを使

用して刺激提示することもできる。

4. 他のツールとの比較

現在よく用いられる開発ツールが現実の研究・教育の場面で与えるコストを比較したのが表1である。一般的なコンピュータ言語をそのまま利用すると、速度面や汎用性で利点があるが、手軽さや心理物理学での使用に難がある。Psychlopsはこれに心理物理学実験に向けた機能を追加し、利用法を簡略化したものと位置づけることができる。

Cambridge Research Systems社によるVisage・VSGなど専用機器は、視覚研究のために非常に優れた機能を持っているものの、たいへん高額であり、少なくとも学生の一人一人に準備するには非現実的な価格設定といえる。実験プログラムの配布や共有といった面でも困難が伴う。

Denis Pelliらが開発したPsychToolBox¹¹⁻¹³⁾は現在広く普及しており、Matlabの機能や豊富な資産蓄積を利用することができる。しかし、Matlabのアカウント管理・維持には金銭面だけでなく事務的にもかなりのコストがかかる。GUI環境を備え、学習が容易なPsyscopeやPsykinematix¹⁴⁾などの実験ツールも存在するが、これらは開発の柔軟性に問題点を抱え、価格やアカウント管理に関してはMatLabと同様のコストがかかるものがある。またMac OS Xへの正式な移植がされていないものもあり、これら

表1 Psychlopsと代表的なツールの比較。Psychlopsは、バイナリ生成可能なフリーのプログラム型心理学向けツールという立場に当たる。

開発環境	手軽さ	心理学向け	フリー環境	汎用バイナリ	柔軟性
専用機器 (VSG等)	×	○	×	×	○
標準C/C++	×	×	○	○	○
GLUT等	○	×	○	○	○
Psychlops	○	○	○	○	○
GUIツール(Ψkinematix等)	○	○	×	×	×○
Psychtoolbox	○	○	× ²⁾	×	○
PsychPy / VisionEgg	○	○	○	×	○

1) ソフトウェアによって柔軟性の程度は異なる

2) Psychtoolbox3よりフリー環境のGNU / Octaveに一部対応

の環境では実験環境の維持も難しい。

一方で、最近になって PsychoPy¹⁵⁾ や VisionEgg¹⁶⁾ といった、フリーのツールも発表された。これらのツールは Psychlops と共通するコンセプトを持ち、OpenGL を用いて Windows, MacOSX どちらでも動作し、刺激表示に特化している。しかし、これらのツールを駆動させるためには Python と呼ばれるスクリプト言語を習得する必要がある。この言語は欧米では広まりつつあるが、現状の日本における一般的な情報処理の授業においてはほとんど取り扱いがない。学習を始めるときに必要なサンプルデモの蓄積や日本における情報も十分とはいえず、総じて学習コストは高い。また、これらのツールは、Python 言語の特性上、縞パターンやランダムドットなど刺激の要素をクラスとしてカプセル化することが強く推奨され、これらのクラスの使い勝手は開発陣の嗜好に大きく左右される。PsychoPy ではオフスクリーン（ビットマップ）の各画素を直接操作する命令がまだ実装されていない。これに対して、Psychlops ではビットマップの描画機能を備え外部 C ライブラリとの結合が可能であるなど、より直接的な下位の命令を扱えるので、ユーザのそれぞれが持つ目的に最適な構成を作ることが可能である。さらに、日本語マニュアルが用意されており、習得時に大きく助けとなる実際に利用可能な公開デモ・プログラムも豊富に用意されている。

5. 入手方法と必要環境

現在 Psychlops は “Psychlops Wiki” (<http://psychlops.l.u-tokyo.ac.jp>) で入手できる。このサイトでは、ライブラリ本体の配布のほか、支援ツールや技術情報の提供、フィードバックの受付を行っている。各種検索サイトや視覚科学データベース “Visiome platform” (<http://platform.visiome.neuroinf.jp/>) において “Psychlops” で検索することでも作成者のホームページや動作例をはじめとする各種情報にアクセスできる。これらのサイトでは、2009年2月現在で Mac OSX (10.4, 10.5), Windows (2000, XP,

Vista) に対応した v1.30 のパッケージがダウンロード可能となっている。利用許諾は LGPL (GNU Lesser General Public License; <http://www.gnu.org/copyleft/lesser.html>) とほぼ同等の条項を採用しており、オープンソースソフトウェアとして公開されている。利用者は制限なく自由にライブラリを利用して実行ファイルを配布することができる。また、利用者はライブラリ自体も自由に改変し公開することが可能である。

Psychlops は現在一般向けに販売されているほとんどのパーソナルコンピュータで動作するが、より高速に描画を行うためにはなるべく新しい高性能のビデオボードをそなえたパソコンであることが望ましい。

6. 終わりに

本稿では心理物理学実験向けのライブラリ Psychlops の概要についてまとめた。心理物理学実験向けのライブラリとして必要な基本部分に関して、Psychlops は少なくとも他のツールと同等の性能を持っている。しかし、現段階での Psychlops には、既存の他の実験用ツールに比べて未成熟な点も多い。特に開発の際のユーザインターフェースは C++ の開発環境に依存しており、初学者には敷居が高いかもしれない。これらの点を改善することは今後の努力目標であるが、これにはある程度の規模を持ったユーザグループの形成が不可欠である。Visiome や PsychlopsWiki などの場でよりユーザ視点に立った議論が行われれば、Psychlops はより利便性の高いソフトウェアへ発展することができる。このようなユーザからのフィードバックと開発の健全なサイクルの形成と、それによるソフトウェアの成熟が今後の重要な課題であると考えられる。

文献

- 1) 堺 浩之, 白井支朗: 視覚研究のデジタル・アーカイブ: Visiome platform. *Vision*. 18, 29–32, 2006

- 2) D. Shreiner, M. Woo, J. Neider and T. Davis: OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 2 (5th Edition). Addison-Wesley Professional, Reading, 2005.
- 3) G. W. Meyers and D.P. Greenberg: Perceptual color spaces for computer graphics. *Computer Graphics*, **14**, 254–261, 1980.
- 4) H. Stanislaw and L. A. Olzak: Parametric methods for gamma and inverse gamma correction, with extensions to halftoning. *Behavior Research Methods, Instruments, & Computers*, **22**, 402–408, 1990.
- 5) W. B. Cowan: An inexpensive scheme for calibration of a colour monitor in terms of CIE standard coordinates. *Computer Graphics*, **17**, 315–321, 1983.
- 6) R. S. Berns: Methods for characterizing CRT displays. *Displays*, **16**, 173–182, 1996.
- 7) 番 浩志, 山本洋紀, 江島義道: Mcalibrator: MATLAB 言語を用いたディスプレイ較正のための統合型 GUI ソフトウェアの開発. *基礎心理学研究*, **24**, 149–161, 2006.
- 8) C. L. Liu and J. Layland: Scheduling algorithms for multiprogramming in a hard real-time environment. *Journal of the ACM*, **20**, 46–61, 1973
- 9) G. C. Buttazzo: Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications. Springer, Berlin, 2005.
- 10) M. Frigo and S. G. Johnson: FFTW: an adaptive software architecture for the FFT. Proceedings of international conference acoustics. *Speech, and Signal Processing*, **3**, 1381–1384, 1998.
- 11) D. H. Brainard: The psychophysics toolbox. *Spatial Vision*, **10**, 433–436, 1997.
- 12) D. G. Pelli: The videotoolbox software for visual psychophysics: transforming numbers into movies. *Spatial Vision*, **10**, 437–442, 1997.
- 13) M. Kleiner, D. Brainard and D. Pelli: What's new in Psychtoolbox-3? *Perception*, **36** Supplement, 2007.
- 14) W. H. A. Beaudot: Psykinematix: A new psychophysical tool for investigating visual impairment due to neural dysfunctions. *Vision*, **21**, 19–32, 2009.
- 15) J. W. Peirce: PsychoPy—Psychophysics software in Python. *Journal of Neuroscience Methods*, **162**, 8–13, 2007.
- 16) A. D. Straw: Vision Egg: an open-source library for realtime visual stimulus generation. *Frontiers in Neuroinformatics*, **2:4**. doi:10.3389/neuro.11.004.2008, 2008.