

MATLABとPsychtoolboxを用いたプログラミングの基礎I： 日本語フォントの表示とガウス関数の計算

前原 吾朗*・実吉 綾子**

*神奈川大学 人間科学部

〒221-8686 神奈川県横浜市神奈川区六角橋3-27-1

**帝京大学 文学部心理学科

〒192-0395 東京都八王子市大塚359

goro@kanagawa-u.ac.jp

1. はじめに

2017年の視覚学会冬期大会の前日（1月17日）に、日本視覚学会若手の会が主催して視覚実験プログラムワークショップが開催された。このワークショップには2つのコース（PsychtoolboxとPsychlops）が用意されており、著者らがPsychtoolboxコースの講師を担当した。Psychtoolboxとは、MATLABで動作する心理学実験用の関数セットで、心理学実験プログラミングに有用な関数が多く用意されている¹⁻³⁾。特に重要なのは、ディスプレイの画面書き換えと視覚刺激呈示との同期を容易に行える点である。こうしたプログラムを自身で一から作成しようとすると、ビデオカードの制御に関する高度なプログラミング技術が必要となってくる。そうした技術的な問題をPsychtoolboxは解決し、心理学実験プログラミングにかかる時間を大幅に短縮してくれる。

著者らのワークショップでは、①視覚探索実験、②日本語フォントを使用した文字列の呈示、③コントラスト弁別実験、④ボイストリガーについて扱った。また、参加者からの質問に答えるかたちで、⑤ガウス関数の作成についても説明した。このうち①視覚探索実験と③コントラスト弁別実験では、具体的な実験プログラムを例示することで、画面書き換えと同期した刺激呈示、キーボードを使った反応の取得、

テキストファイルへの反応記録といった心理学実験に重要な要素を学ぶことを目標とした。これらの実験プログラムと④ボイストリガーのプログラムについては、実吉・前原⁴⁾の著書『はじめよう実験心理学：MATLABとPsychtoolboxを使って』において詳細に説明されている。本講義論文では、ワークショップに際して参加者から要望のあった②日本語フォントを使用した文字列の呈示と⑥ガウス関数の作成について解説する。

2. 日本語フォントを使用した文字列の呈示

付録1にワークショップで例示したプログラムTextDemoJapanese.mを示した。このプログラムはPsychtoolboxウェブサイトのチュートリアルにあるBasic Text Demo (<http://peterscarfe.com/textdemo.html>)を、Psychtoolbox3のデモフォルダ(PsychDemos)内にあるDrawHighQualityUnicodeTextDemo.mを参考に日本語の文字列が表示されるように改訂したものである。TextDemoJapanese.mを実行すると「Hello World」が3つと「こんにちは」が画面に表示される。使用PCがMACの場合には使用フォントを変更する必要があるかもしれない(詳細は次段落を参照)。

元のデモプログラムに追加した部分は、「%日本語を表示する」のコメントの下の4行である。まず、1行目のScreen('TextFont', window, 'MS Mincho');において、テキストフォントをMS明朝に設定している。Macで動

作しなかった場合には、'MS Mincho'の代わりに'Hiragino Mincho Pro'を入力し、フォントをヒラギノ明朝Proに設定しておく。Screen('TextFont')の2つめのパラメータwindowは文字列を表示するウィンドウへのポインタで、3つめのパラメータは設定するフォントの名称となっている。

2行目では、MATLAB標準関数のnative2unicode()を使って日本語の文字列をユニコードに変換する。文字はフォントごとに異なる番号が割り振られて管理されているが、世界中で使われている文字を共通の番号で管理するのがユニコードである。このユニコードへの変換を行わなければ、日本語フォントの文字列は正しく表示されない。native2unicode()へ対しては、変換する文字列を第一パラメータとして、その文字列の符号化形式（ここでは'Shift_JIS'）を第二パラメータとして渡す必要がある。

3行目のMATLAB標準関数double()がユニコード文字表現をdouble形式の変数に変換している。Psychtoolbox関数のDrawFormattedText()やScreen('DrawText')はユニコード表現をそのまま受け取ることができないので、double形式に変えなくてはならない。このdouble()が変換するのは変数の形式のみで、文字の番号自体は変化しない。ここでは、ユニコード文字表現の文字列textをdouble形式に変換し、変数unicodetextに入れている。

最後にDrawFormattedText()を使って文字列の描画を行う。DrawFormattedText()の第一パラメータには描画を行うウィンドウのポインタ、第二パラメータには文字列、第三パラメータと第四パラメータには文字列表示のx座標とy座標、第五パラメータには文字の色をそれぞれ入力する。x座標とy座標に'center'を入力すると、それぞれ文字列を横中央揃えと縦中央揃えにして表示することができる。Basic Text DemoではDrawFormattedText()が用いられていたが、代わりにScreen('DrawText')を使用しても問題はない。ただし、Screen('DrawText')では座標を'center'に指定することはできない。

ついでながら、Screen('DrawText')で指定する描画位置は文字列の左上の座標となっている。文字列の中心座標で描画位置を指定したい場合には、Screen('TextBounds')で文字列を囲む四角形を取得する必要がある。こうした処理に関しては、DrawFormattedText()がScreen('TextBounds')とScreen('DrawText')を使って中央揃えの文字を描画しているので、そのコードを読むと参考になる。

3. ガウス関数の計算

ワークショップ参加者の一人から、ガウス関数に従って輝度が増える視覚刺激の計算方法についての質問があったので、付録2のGenerateGaussian()を例に解説を行った。

この関数は、第一パラメータにガウス関数の分散を、第二パラメータに画像サイズを受け取るように作られている。それぞれ単位はピクセルで、例えばgaussian=GenerateGaussian(20, 200);を実行した場合、200×200ピクセルの画像の中心に標準偏差20ピクセルのガウス関数が描かれる。image processing toolbox (Mathworks社提供の有料toolbox)を利用可能なら、imshow(gaussian);を実行すれば画像を確認することができる。Psychtoolbox関数のScreen()を使用して表示することもできる（詳細は実吉・前原⁴⁾を参照）。

GenerateGaussian()における最初の命令はMATLAB関数のmeshgrid()である。図1aに[xx yy]=meshgrid(1:10, 1:10);の出力を例示した。meshgrid()を実行すると渡されたパラメータの範囲で数値が変動する行列が作成される（第一パラメータはx座標の範囲を、第二パラメータはy座標の範囲を決定する）。このとき、xxにはx座標の値が、yyにはy座標の値が入る（図1a）。例えば、xx(1,5)は5だが、yy(1,5)は1となる。こうした行列は四角形グリッドと呼ばれている。GenerateGaussian()では[x, y]=meshgrid(1: imageSize, 1: imageSize);が実行されているので、imageSizeが200なら数値範囲が1～200の四角形グリッドx, yが作成される。

a) `[xx yy] = meshgrid(1:10, 1:10);`を実行

変数 xx		列									
		1	2	3	4	5	6	7	8	9	10
1		1	2	3	4	5	6	7	8	9	10
2		1	2	3	4	5	6	7	8	9	10
3		1	2	3	4	5	6	7	8	9	10
4		1	2	3	4	5	6	7	8	9	10
5		1	2	3	4	5	6	7	8	9	10
6		1	2	3	4	5	6	7	8	9	10
7		1	2	3	4	5	6	7	8	9	10
8		1	2	3	4	5	6	7	8	9	10
9		1	2	3	4	5	6	7	8	9	10
10		1	2	3	4	5	6	7	8	9	10

変数 yy		列									
		1	2	3	4	5	6	7	8	9	10
1		1	1	1	1	1	1	1	1	1	1
2		2	2	2	2	2	2	2	2	2	2
3		3	3	3	3	3	3	3	3	3	3
4		4	4	4	4	4	4	4	4	4	4
5		5	5	5	5	5	5	5	5	5	5
6		6	6	6	6	6	6	6	6	6	6
7		7	7	7	7	7	7	7	7	7	7
8		8	8	8	8	8	8	8	8	8	8
9		9	9	9	9	9	9	9	9	9	9
10		10	10	10	10	10	10	10	10	10	10

b) `xx = xx-(10+1)/2; yy = yy-(10+1)/2;`を実行

変数 xx		列									
		1	2	3	4	5	6	7	8	9	10
1		-4.5	-3.5	-2.5	-1.5	-0.5	0.5	1.5	2.5	3.5	4.5
2		-4.5	-3.5	-2.5	-1.5	-0.5	0.5	1.5	2.5	3.5	4.5
3		-4.5	-3.5	-2.5	-1.5	-0.5	0.5	1.5	2.5	3.5	4.5
4		-4.5	-3.5	-2.5	-1.5	-0.5	0.5	1.5	2.5	3.5	4.5
5		-4.5	-3.5	-2.5	-1.5	-0.5	0.5	1.5	2.5	3.5	4.5
6		-4.5	-3.5	-2.5	-1.5	-0.5	0.5	1.5	2.5	3.5	4.5
7		-4.5	-3.5	-2.5	-1.5	-0.5	0.5	1.5	2.5	3.5	4.5
8		-4.5	-3.5	-2.5	-1.5	-0.5	0.5	1.5	2.5	3.5	4.5
9		-4.5	-3.5	-2.5	-1.5	-0.5	0.5	1.5	2.5	3.5	4.5
10		-4.5	-3.5	-2.5	-1.5	-0.5	0.5	1.5	2.5	3.5	4.5

変数 yy		列									
		1	2	3	4	5	6	7	8	9	10
1		-4.5	-4.5	-4.5	-4.5	-4.5	-4.5	-4.5	-4.5	-4.5	-4.5
2		-3.5	-3.5	-3.5	-3.5	-3.5	-3.5	-3.5	-3.5	-3.5	-3.5
3		-2.5	-2.5	-2.5	-2.5	-2.5	-2.5	-2.5	-2.5	-2.5	-2.5
4		-1.5	-1.5	-1.5	-1.5	-1.5	-1.5	-1.5	-1.5	-1.5	-1.5
5		-0.5	-0.5	-0.5	-0.5	-0.5	-0.5	-0.5	-0.5	-0.5	-0.5
6		0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
7		1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5
8		2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5
9		3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5
10		4.5	4.5	4.5	4.5	4.5	4.5	4.5	4.5	4.5	4.5

図1 四角形グリッドの計算.

a) `meshgrid()`の出力例. b) 行列中心を0とした座標への変換.

次に、`x=x-(imageSize+1)/2;`と `y=y-(imageSize+1)/2;`を実行し、行列の中心を0とした座標への変換を行う。前段落の `xx`と `yy`を例にすると、この計算結果は図1b)のようになる。行列中心が0の場合の `x`座標を `xx`が、`y`座標を `yy`が表現していることが見て取れる。これと同様の変換が、`GenerateGaussian()`の四角形グリッド `x,y`に対しても行われる。

座標の変換が完了したら、`radius=sqrt(x.*x + y.*y);`を実行する。この命令は「`x`座標の二乗と `y`座標の二乗を足した値の平方根」という

半径の計算を行い、行列中心からの距離を変数 `radius`に出力する。 `imageSize`が200の場合、`radius`は200×200の行列で、行列中心に近いほど値が小さく(0に近く)、遠いほど値が大きくなる。

ガウス関数は以下の式で定義される。

$$f(x) = \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}$$

μ は平均値(ガウス関数の値が最大となる座標)で、 σ は標準偏差である。`generateGaussian()`

では平均値は0, 標準偏差は変数sdとするので, $\text{gaussian} = \exp(-(\text{radius}^2)/(2*\text{sd}^2))$; を実行してガウス関数の計算を行う. 変数radiusをガウス関数の入力とすることで, 変数gaussianはガウス関数に従って値が変動するimageSize×imageSizeの行列となる.

4. む す び

MATLABを使用することの利点として, 変数・行列の操作や基本的な計算に有用な関数が数多く用意されていることがあげられる. meshgrid() もそのひとつで, 筆者はMATLABに触れることで四角形グリッドを用いた計算手法を知った. このやり方はサイン波グレーティングを作成する際にも便利なので, 別の機会があれば解説したい.

Psychtoolboxの開発は20年におよぶ歴史を持つが, 現在も発展中である. ウェブページは常に更新されており, アップデートされた関数を用いたチュートリアルも最近公開された. 古いデモプログラムが動かないというのがPsychtoolboxの問題のひとつだと筆者は考えていたのだが, 新しいチュートリアルにはわかり

やすいデモが充実しており, この短所は克服されたといってよいだろう. 今後の発展が期待させるPsychtoolboxに関する情報をこれからも追っていく所存である.

謝 辞 この度, 視覚実験プログラムワークショップを企画し, 講師にお誘いいただきました視覚学会若手の会の皆様に感謝いたします.

文 献

- 1) D. Brainard: The psychophysics toolbox. *Spatial Vision*, **10**, 433-436, 1997.
- 2) M. Kleiner, D. Brainard and D. Pelli: What's new in Psychtoolbox-3? Paper presented at the European Conference on Visual Perception, Arezzo, Italy, 2007.
- 3) D. Pelli: The VideoToolbox software for visual psychophysics: Transforming numbers into movies. *Spatial Vision*, **10**, 437-442, 1997.
- 4) 実吉綾子, 前原吾朗: はじめよう実験心理学: MATLABとPsychtoolboxを使って. 勁草書房, 2015.

付録1 TextDemoJapanese.m

```
% Clear the workspace and the screen
close all;
clearvars;
sca

% Here we call some default settings for setting up Psychtoolbox
PsychDefaultSetup(2);

% Get the screen numbers
screens = Screen('Screens');

% Select the external screen if it is present, else revert to the native screen
screenNumber = max(screens);

% Define black, white and grey
black = BlackIndex(screenNumber);
white = WhiteIndex(screenNumber);
grey = white / 2;

% Open an on screen window and color it grey
[window, windowRect] = PsychImaging('OpenWindow', screenNumber, grey);

% Set the blend function for the screen
Screen('BlendFunction', window, 'GL_SRC_ALPHA', 'GL_ONE_MINUS_SRC_ALPHA');

% Get the size of the on screen window in pixels
% For help see: Screen WindowSize?
[screenXpixels, screenYpixels] = Screen('WindowSize', window);

% Get the centre coordinate of the window in pixels
% For help see: help RectCenter
[xCenter, yCenter] = RectCenter(windowRect);

% Draw text in the upper portion of the screen with the default font in red
Screen('TextSize', window, 70);
DrawFormattedText(window, 'Hello World', 'center', screenYpixels * 0.25, [1 0 0]);

% Draw text in the middle of the screen in Courier in white
Screen('TextSize', window, 80);
```

```

Screen('TextFont', window, 'Courier');
DrawFormattedText(window, 'Hello World', 'center', 'center', white);

% Draw text in the bottom of the screen in Times in blue
Screen('TextSize', window, 90);
Screen('TextFont', window, 'Times');
DrawFormattedText(window, 'Hello World', 'center', screenYpixels * 0.75, [0 0 1]);

% 日本語を表示する
Screen('TextFont', window, 'MS Mincho');
text = native2unicode(' こんにちは', 'Shift_JIS');
unicodetext = double(text);
DrawFormattedText(window, unicodetext, 50, 'center', white);

% Flip to the screen
Screen('Flip', window);

% Now we have drawn to the screen we wait for a keyboard button press (anykey) to terminate the
demo
KbStrokeWait;

% Clear the screen
sca;

```

付録2 GenerateGaussian.m

```

function gaussian = GenerateGaussian(sd, imageSize)

% sd: pixel
% imageSize: pixel

[x, y] = meshgrid(1:imageSize, 1:imageSize);
x = x-(imageSize+1)/2; % cartesian coordinates
y = y-(imageSize+1)/2; % cartesian coordinates
radius = sqrt(x.*x + y.*y); % radial polar coordinates

gaussian = exp(-(radius.^2)/(2*sd^2)); % Gaussian Stimulus: 0 to 1

```