

情報学群実験第 3C/3i 実験レポート 第 4 回
視覚情報処理とデータ解析に関する実験

1250373 溝口 洋熙*
Group 10

June 21st, 2023

概要

今回の実験では、眼球運動測定実験、行動実験、脳波測定実験をする。眼球運動測定実験では、顔画像の印象に関する判断についての眼球運動と、対象を自由に見るときの眼球運動を測定する。カスケード現象より選好する顔画像を選択する直前の視線の向き、またポスターに対する注視する箇所について明らかになった。行動実験では、多数ある妨害刺激中に目標刺激があるか否かを判断し、刺激数と探索時間の関係を確認する。結果に対して回帰直線の傾きを求め、刺激の種類によって、傾きの異なることが、統計的仮説検定を用いて明らかとなった。脳波測定実験では、BMI 装置を用いて脳波を読み取り、指定したターゲット刺激が表示された脳波は、そうでない場合の脳波に比べて電位の振れがより目立った。

目次

第 1 章	顔画像の印象に関する判断についての眼球運動	1
1.1	実験の目的	1
1.2	実験の方法と考え方	1
1.3	実験の結果	3
1.4	考察	3
1.5	結論	3
第 2 章	対象を自由に見るときの眼球運動	4
2.1	実験の目的	4
2.2	実験の方法と考え方	4
2.3	実験の結果	5
2.4	考察	5
2.5	結論	5
第 3 章	行動実験と統計的仮設検定	6
3.1	実験の目的	6
3.2	実験の方法と考え方	6
3.3	実験の結果	9
3.4	考察	10
3.5	結論	10
第 4 章	脳波測定実験	11
4.1	実験の目的	11
4.2	実験の方法と考え方	11
4.3	実験の結果	12
4.4	考察	13
4.5	結論	13
第 5 章	関連語句	14
5.1	脳情報デコーディング	14
5.2	視線入力インタフェース	14
5.3	顕著性マップ	14
5.4	分散分析	15
参考文献		16
付録		17
A	顔画像の印象に関する判断についての眼球運動 (June 1st, 2023)	17
B	行動実験と統計的仮設検定 (June 5th and 8th, 2023)	17

第1章

顔画像の印象に関する判断についての眼球運動

1.1 実験の目的

■カスケード現象 嗜好判断が下される際の眼球運動に注目した研究がある。複数の対象を比較しながら好みのものを選ぶ際には、選好する刺激に対して頻繁に視線を向けること^{*}が知られており、次のような実験が行われた。左右に並んだ顔写真を呈示し、「どちらが魅力的か」を判断してもらう。顔画像を呈示してから、判断するまでの眼球運動を分析すると、画像呈示直後は両者の顔画像を、およそ 50% ずつの割合で見比べるが、その後、選好する顔画像を見る確率が次第に増加し、80% を超えたところで長く見た顔写真を「魅力的だ」を決定する。本実験中の「片方の顔画像を見る確率が次第に増加する」現象を、視覚のカスケード現象（以下、カスケード現象）と呼ぶ。Shimojo *et al.* (2003) は、一方の顔画像に対して注視時間が長くなることによりもう一方の刺激を精査する時間が短くなることによって生じると解釈した。

[1, p.202], [2]

■目的 本実験では、意思決定直前の眼球運動を計測をする。画像が呈示された瞬間から、判断するまでの時間に対して、魅力的であると判断した画像を注視した時刻に 1, そうでない時刻に 0 を割り当て、試行回数 20 回に対しての割合を求め、カスケード現象が実際に起きているか確認する。カスケード現象の有無と、理想データと実験データの差異について考察する。

1.2 実験の方法と考え方

■実験装置 眼球運動の計測には、Eye Link II (SR Research 社) を用いる。刺激呈示用装置として、汎用コンピュータと液晶ディスプレイを利用する。データ分析するソフトウェアには MATLAB[®]を用いる。

表 1-1: 実験装置（顔画像の印象に関する判断についての眼球運動）

眼球運動計測器	Eye Link II	SR Research 社 (角膜反射法)
刺激呈示コンピュータ	プロセッサ	Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz 3.40 GHz
	メモリ	4GB
	OS	Microsoft Windows 7 Professional Service Pack 1
データ分析コンピュータ	コンピュータ	MacBook Air 2022 (Apple 社) MLY13J/A
	プロセッサ	Apple Silicon M2 8 コア CPU, 8 コア GPU
	メモリ	8GB
	OS	macOS 13.4
MATLAB [®]		R2023a - academic use (Update1 9.14.02239454)
		64-bit (maci64) March 30, 2023

*選好注視と呼ばれている。

■眼球運動の計測 顔の印象に関して判断するときの、眼球運動を計測する。実験参加者は、学生の20代男性1名である。Eye Link IIのアイカメラを装着し、キャリブレーション*およびバリデーション†を行う。

- [1] 注視点を表示する。
- [2] 2つの顔画像を表示する。顔画像の例を図1-2に示す。
- [3] 魅力的であると判断した顔画像を、キー操作により選択する。

[1]から[3]を20回繰り返す。



図1-1: 実験の様子

■実験データの分析 実験データをCSVファイルで出力したものに対して分析する。出力されたCSVファイルと、データ処理過程で作成するexpos行列のIndexを以下に示す。

exp4i_g0310.csv	g0310.asc_TRIAL_N.csv (N = 1, 2, ..., 20)	expos
-----------------	-------------------------------------------	-------

1列目 試行回数	1列目 時間 (ms)	1列目 時間
2列目 選択画像	2列目 測定する目の情報 (未使用)	2列目 画面の状態
◊ 100: 左を選択	3列目 画面の状態	3列目 左右どちらを見ているか
◊ 102: 右を選択	◊ 0: 画面表示なし	4列目 選択した方を見ているか
	◊ 1: 注視点	5列目 x 座標
	◊ 2: 画像表示	
	4列目 左右注視位置 x 座標	
	5列目 左注視位置 y 座標 (未使用)	
	6列目 左目瞳孔径 (未使用)	

■欠損値 本実験機Eye Link IIは時間周波数500Hzでデータを取得している。この場合、隣接するサンプル間の時間感覚は2msとなる。眼球の向きが計測できない場合（まばたきなど）、その部分のデータは記録されたため欠損値として扱う。

■データ処理の手順 本実験では、表示された顔画像に対して魅力的であると判断した時刻から1秒さかのぼった時刻を-1000msとし、選択した顔画像を見ている場合を1、そうでない場合を0として、その合計を欠損値を除いた試行回数で割り、0msから500msの各時刻に対して、魅力的であると判断した画像を注視した割合を算出する。



(a) 左に表示 (b) 右に表示

図1-2: 表示される顔画像（例）

1. exp4i_g0310.csvをreadmatrix関数を用いて読み込む。
2. 試行回数分のループ中に次の処理を行う。ループ変数をkとする。
 - i. 各試行結果をreadmatrix関数を用いて読み込む(data)。
 - ii. dataの行サイズ(sizeR)、列サイズ(sizeC)に対して、欠損値がない場合の理想行サイズ(ms)を求める。
 - iii. exposをms行5列で初期化する。初期値を0にするため、zeros関数を用いる。
 - iv. dataとexposに対して判断した瞬間を時刻0とするため、dataとexposそれぞれに対して、全行1列と1行1列の差をとる。
 - v. デフォルトで欠損値とするため、exposの2行目から5行目へ-1を代入する。
 - vi. exposとdataの1列目に対して欠損値を検出するため、ismember関数を用いる。この関数は、第1引数にあ

*被験者の中心点を、実験機に記憶させる手続き。

†キャリブレーションが正常に完了したことを確認する手続き

る行が第2引数にあれば1, そうでなければ0を返す。この真理行列をbinとする。

- vii. exposの2列目にdataの3列目, exposの5列目にdataの4列目を代入する。このとき, binが1の行のみ exposへ代入する。
 - viii. 注視点を見ているとき, 左目x座標の平均値(x_ave)に対して, 左目x座標がx_ave以下の場合は「左を見ている」, x_aveより大きい場合は「右を見ている」とし, expos3列目に100または102を代入する。欠損値に対しては処理を行わない。
 - ix. 最終的に魅力的であると判断した顔画像を注視している場合は1, そうでない場合は0をexposの4列目に代入する。この処理は, 画面状態が0と欠損値に対しては行わない。
 - x. 500行20列で初期化済みのmergeC行列に対して, 全行k列にexposの1から500行目(判断前1秒間)のデータを格納する。
3. 各時刻(2msごと)を格納するratio変数を初期化する。
4. 魅力的であると判断した顔画像を注視している割合を, 欠損値の試行を除いた試行回数で割ることにより求める。
5. 時間軸に対して, ratioをplot関数を用いて, グラフを描画する(src.A-1).
►p.17
6. 考察のために, キーを押下し選択する1200msからのデータと, そのデータに対してpolyfit関数とpolyval関数を用いて, x軸time, y軸ratioの3次近似多項式を取り, グラフに描画する(src.A-2).
►p.17

1.3 実験の結果

実験結果を図1-3, 図1-4に示す。

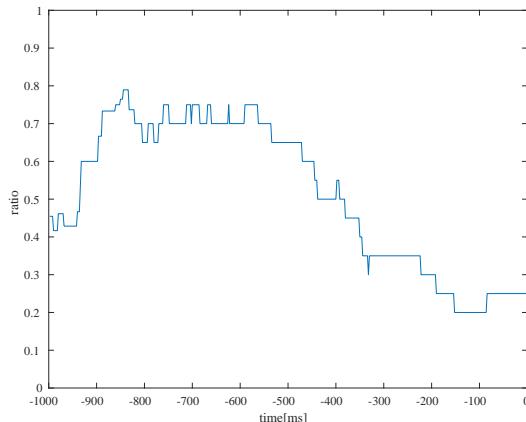


図1-3: 1s前からの結果

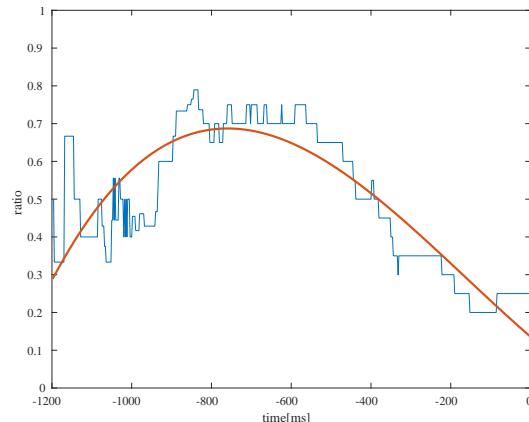


図1-4: 1.2s前からの結果と近似多項式曲線

1.4 考察

実験結果ではカスケード現象を確認できなかった。原因として, 魅力的であると判断した後に少し時間を置いてからキーを押下し選択していることが考えられる。図1-4より, 判断前800msにピークを迎えており, この時刻が魅力的であると判断した時刻であろう。その前提を真とすると, 判断前1200msから800msで, 割合は増加傾向にあり, これがカスケード現象と考えられる。

1.5 結論

本実験を通して, カスケード現象を正しく検出するためには, 魅力的であると判断した瞬間の選択行為が必要であり, 魅力的であると判断してから選択行為の間に時間があると, カスケード現象を正しく検出できないことが分かった。

第2章

対象を自由に見るときの眼球運動

2.1 実験の目的

我々は、ポスターなどの写真を注視するとき、どこをよく見るだろうか。どこを注視しているか追跡する技術を「アイトラッキング技術」と呼ぶが、この技術はマーケティングの分野にも応用されている。金融機関店舗における利用者の注視行動の調査[3]では、注視したとされるものは印象に残ることが分かっている。本実験では、人の顔、指差し、文字が含まれた刺激画像に対して眼球運動を計測し、刺激画像中のどこを注視しているかを定量化する。

2.2 実験の方法と考え方

■実験装置 今回は、アイトラッキングするための装置として Tobii 社の Tobii Grass 3 を用いる。データ分析をするコンピュータを表 2-1 に示す。

表 2-1: 実験装置（対象を自由に見るときの眼球運動）

アイトラッキング装置	Tobii Grass 3	Tobii 社（角膜反射法）
データ分析コンピュータ	コンピュータ	64 ビット オペレーティング システム, x64 ベース プロセッサ
	プロセッサ	Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz 2.59 GHz
	メモリ	32.0 GB (31.6 GB 使用可能)
	OS	Windows 10 Pro (21H2)
	OS Build	19044.2251
	Experience	Windows Feature Experience Pack 120.2212.4180.0

注視する対象は図 2-1 で、本実験の参加者は、学生の 20 代男性 1 名である。実験手続を以下に示す。

- 1 Tobii Grass 3 のメガネを装着する。
- 2 呈示されるマーカを追視し、視線方向のキャリブレーションを行う。
- 3 刺激（図 2-1）を自由に観察したときの Gaze Plot, Heat Map を作成する。

■Gaze Plot と Heat Map 数値と円の大きさで、注視た順番と注視た時間を表現するグラフである。数値は、注視した順番を表し、円の大きさは注視した時間を表す。円の大小と、注視時間は比例関係にある。

■Heat Map 注視時間が長い箇所を赤く表示する図である。



図 2-1: 注視画像

2.3 実験の結果

作成した Gaze Plot および Heat Map を図 2-2 に示す。注視箇所について、差された指の指先と顔の中央、文字に集中していることが分かる。注視した順番について、各番号の偏りは見られなかった。



図 2-2: 対象を自由に見るときの眼球運動 実験結果

2.4 考察

我々はポスターを見るときに、文字情報や被写体が印象に残る。結果より、文字や被写体の顔部分をより多く注視しているからではないだろうか。注視した順番については、全体的に偏りは見られないものの、文字列に対しては比較的小さな数値が出力されている。我々は、まず文字から情報を得ようとするのではないだろうか。

2.5 結論

本実験を通して、刺激画像の注視箇所と我々の印象に残っている点の因果を考察できた。しかし、差された指の指先も注目すべき注視箇所だが、なぜ指先が多く注視されているのかの原因は分からなかった。

第3章

行動実験と統計的仮設検定

3.1 実験の目的

本実験では、探索時間を指標とした視覚探索課題の心理物理実験をする。心理物理実験とは、物理的世界に属する刺激強度と、心理的世界に属する感覚強度の数学的関数関係を明らかにする実験である[4]。今回は、視覚探索課題に対して心理物理実験をする。視覚探索課題とは、複数の刺激項目に対して、あらかじめ決められた目標項目があるか否かを、実験参加者が判断する課題である[5]。複数の刺激「妨害刺激」の中に、1つだけ異なる刺激「目標刺激」が存在する場合と、そうでない場合について、刺激数に対する探索時間を計測する。実験参加者は、目標刺激が妨害刺激の中にあるかないかを、できるだけ早くかつ正確に判断する必要がある。

また、結果に対して統計的仮説検定をする。統計的仮説検定とは、標本を使って母集団に関する統計的な判断を下す方法である[6, p.200]。探索時間が限りなく0になる「ポップアウト」、刺激数と探索時間が比例する「逐次探索（系列探索）」についての研究[7]で、刺激種類と探索時間の関係について言及されている。今回は t 検定を用いて、視覚探索課題における探索非対称性*があるか検討する。仮説は「探索非対称性がある」とし、仮説が正しい場合、目標刺激が異なる2条件に対して回帰直線の傾きに差が生じることを確認する。その結果を踏まえて、刺激の種類と「ポップアウト」、「逐次探索」の関係について考察する。

3.2 実験の方法と考え方

■実験装置 視覚探索課題の心理物理実験には、MATLAB[®]とPsychtoolboxを用いる。Psychtoolboxは、プログラミングを簡便にするツールであり、心理物理学の実験でよく用いられる。データ分析するソフトウェアにはMATLAB[®]とデータを集計するGoogleスプレットシートを用いる。

表3-1: 実験装置（行動実験と統計的仮設検定）

刺激呈示コンピュータ	プロセッサ	Intel(R) Xeon(R) Gold 5218R CPU @ 2.10GHz
	メモリ	11.7GB
	OS	Microsoft Windows 10 Enterprise LTSC 64bit
	MATLAB [®]	R2020b - academic use
データ分析コンピュータ	Psychtoolbox	バージョン不明
	コンピュータ	MacBook Air 2022 (Apple社) MLY13J/A
	プロセッサ	Apple Silicon M2 8コアCPU, 8コアGPU
	メモリ	8GB
	OS	macOS 13.4
	MATLAB [®]	R2023a - academic use (Update1 9.14.02239454) 64-bit (maci64) March 30, 2023

*目標刺激と妨害刺激が交代しただけで、探索時間が非常に異なる現象のこと[8]。

■視覚探索課題の心理物理実験 以下に視覚探索課題の心理物理実験方法を示す。

<実験1> (src.B-1, src.B-4)
▶p.17 ▶p.18

- [1] 注視点が 500ms 表示される。
- [2] 画面に刺激 (刺激対 1 : 図 3-1) が表示される。画面に表示される合計の刺激数は 4, 8, 16 個のいずれかである。円の刺激を C, 円と棒の刺激を CL とする。刺激の大きさは, 40 × 40 pixel とする。
- [3] 目標刺激が妨害刺激の中にある場合はキーボード上「F」, そうでない場合は「J」を押下し [1] に戻る。呈示時間は 5000ms で, それまでに操作がなければ Time over とし, [1] に戻る。

[1] から [3] を 240 回繰り返す。本実験の参加者は学生の 20 代男女 14 名である。

<実験2> (src.B-1, src.B-5)
▶p.17 ▶p.19

ここでは, 表示する刺激対を 2 つ用意して, 同様に実験し, 探索時間を記録する。刺激対 2 (図 3-2) 左の刺激を Star, 右の刺激を Diamond とする。この刺激を選択した理由は, 一方の画像に何らかの情報を附加した画像が, もう一方の画像になるからである。図 3-2 の左図の鋭角を 4 つから 5 つに増やす操作をすると, 右図の刺激を得られ, これは図 3-1 の円に線を附加して得られるもう一方の刺激に習って作成した。本実験の参加者は, 学生の 20 代男性 1 名 (参加者 A) である。

■統計的仮説検定 ここで示す母集団は, 調査の対象となる要素の集合。今回の母集団は全人類である。また, 母集団の一部を抽出した集合を標本と呼び, 標本に対して検定して母集団の性質を推測する。以下の統計的仮説検定に用いるデータは 実験1 のデータである。

帰無仮説 H_0 : 述べたい事柄を否定する仮説。帰無仮説を棄却することで, 述べたい事柄を示す。今回は, 視覚探索課題において「探索非対称性がないこと」が帰無仮説となる。

対立仮説 H_a : 述べたい事柄を支持するような仮説を立てる。

t 分布: 正規分布に従う母集団の平均値, 分散が不明である場合に利用する確率分布。標本数を n , 標本の平均値を \bar{X} , 不偏分散を s^2 とし t 分布は式 (3.1) で定義される。 $n - 1$ の値を自由度と呼び, t 分布は自由度が増すほど正規分布に近くなる [9, p.178 - p.179]。

$$T = \frac{\bar{X} - \mu}{\sqrt{s^2/n}} \quad (3.1)$$

t 検定: t 分布を利用する検定であり, ある値に対して, 「ある値と異なる」ことが言えるかどうかを判断する検定方法。ここで用いるある値を t 値と呼ぶ。今回は自由度を n としたときの t 値を式 (3.2) に定め, 両側検定を用いる。

$$t(n) = \frac{\text{回帰直線傾きの平均}}{\text{回帰直線傾きの標準誤差}} \quad (3.2)$$

有意確率 p : 標本と帰無仮説が矛盾となる目安指標。 p が小さいほど帰無仮説と標本が矛盾している。

有意水準 α : 帰無仮説を棄却する基準となる値。有意確率 p が有意水準を下回ったときに帰無仮説を棄却する。一般的には 5% か 1%。本実験では, 有意水準を 5% に設定する。

- ◊ $p < \alpha$: 有意差があり, 帰無仮説が起こる確率は低く帰無仮説を棄却できる。つまり対立仮説を採択できる。
- ◊ $p > \alpha$: 有意差がなく, 帰無仮説が起こる確率を無視できず, 帰無仮説を棄却できない。

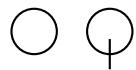


図 3-1: 刺激対 1

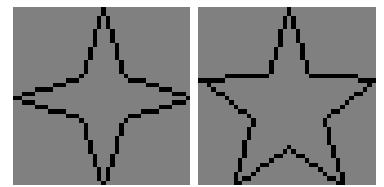


図 3-2: 刺激対 2 (40 × 40 pixel)

標準誤差 SE: 理論上の「標本平均の標準偏差」の大きさ。母集団の統計量の誤差を、標本から母集団の性質を推定した統計量の標準偏差として表したもので、式(3.3)で求める。

$$SE = \frac{\text{標準偏差}}{\sqrt{\text{標本サイズ}}} \quad (3.3)$$

回帰直線: 標本 $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ に対して、

$$(y - y_1)^2 + (y - y_2)^2 + \dots + (y - y_n)^2$$

を最小にするような直線 $y = ax + b$ を回帰直線という。

[6, p.168, p.187, p.200 - p.205]

1. 帰無仮説、対立仮説を立てる。
2. 有意水準 α を設定する。今回は $\alpha = 0.05$ 。
3. 検定統計量 t 値を求める。
4. t 値から有意確率 p 値を求める。
5. 帰無仮説の棄却、採択の判断をする。
 - ◊ $p < \alpha$: 帰無仮説を棄却し、対立仮説を採択する。
 - ◊ $p > \alpha$: 帰無仮説を棄却しない。
6. MATLAB®を用いて、データの平均値と標準誤差の誤差線を出力する (src.B-2).
►p.18

本実験では、目標刺激が C と CL それぞれの目標刺激の有無に対して、探索非対称性がないことを帰無仮説とする。当然、「探索非対称性があること」が対立仮説となる。データの集計には Google スプレットシートを用いる。以下に Google スプレットシート上で用いる関数と、表3-2に集計方法を示す。

AVERAGE(arg) 指定範囲の平均。

VAR.S(arg) 指定範囲の不偏分散。

STDEV.S(arg) 指定範囲の不偏標準偏差（標本標準偏差）。

COUNT(arg) 指定範囲の要素数。

SQRT(arg) 引数の平方根。

TDIST(arg1,arg2,arg3) p 値を出力する。arg1 に t 値を、arg2 に自由度 要素数 - 1 を、arg3 に利用確率（今回は両側を利用するので 2 を代入）を格納する。

SLOPE(arg1,arg2) 回帰直線の傾きを出力する。横座標に arg1、縦座標に arg2 を指定する。

表3-2: Google スプレットシートを用いたデータ集計

参加者番号	反応時間												回帰直線の傾き					
	Target Type: C			Target type: CL														
	Target: YES		Target: NO		Target: YES		Target: NO		C*	CL*	差							
	4	8	16	4	8	16	4	8	16	4	8	16						
NoT	a_1	a_2	a_3										SLOPE(a,NoT) →	m	n	$m - n$		
平均値	AVE															range_s →		
不偏分散	UV															sUV		
標準偏差	SD															sSD		
標準誤差	SE															sSE		

t 値を $t = sAVE/sSD$, p 値を $p = TDIST(t, COUNT(range_s)-1, 2)$ で求め、表3-2を CSV 形式で書き出す。

*目標刺激の種類

次に、MATLAB[®]でデータの平均値と標準誤差の誤差線を出力するスクリプトを記述する。`importdata` 関数を用いて書き出した CSV を読み込む。参加者番号を削除した後に、各統計量を求める。そのデータから平均値と標準誤差の誤差線を `err_data` 関数を用いて描画する。

最後に、読み込んだデータから回帰直線を求める。表 3-2 に示した方法を MATLAB[®]で再現し、`polyfit` 関数と `slope` 関数を用いて回帰直線の傾きと切片を求める。参加者 A のデータを散布図として、回帰直線と同時に描画する (src.B-3). ▶p.18

3.3 実験の結果

<実験 1>で得られた実験参加者の平均値と標準誤差の誤差線を図 3-4、参加者 A の結果に対する平均値と回帰直線を図 3-5 に示す。グラフ作成のもととなったデータを表 3-3、表 3-4 に示す。参加者 A は「参加者番号」列に明記した。

<実験 2>で得られた、刺激対 (図 3-1、図 3-2) の刺激数に対する探索時間を示したグラフを図 3-6 に示す。

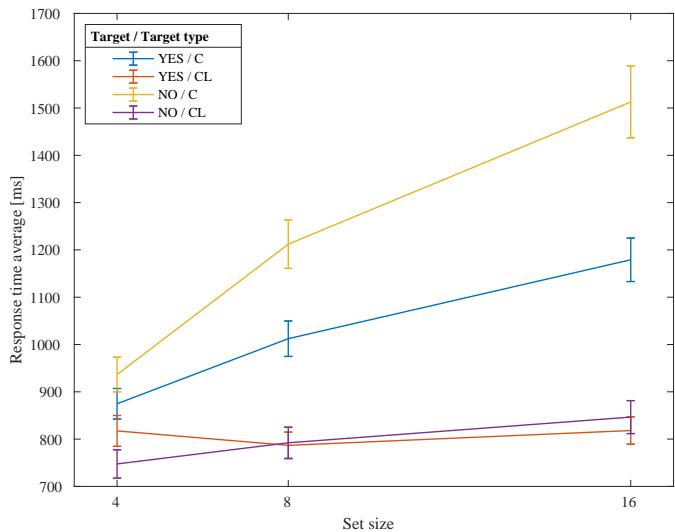


図 3-4: 平均値と誤差線

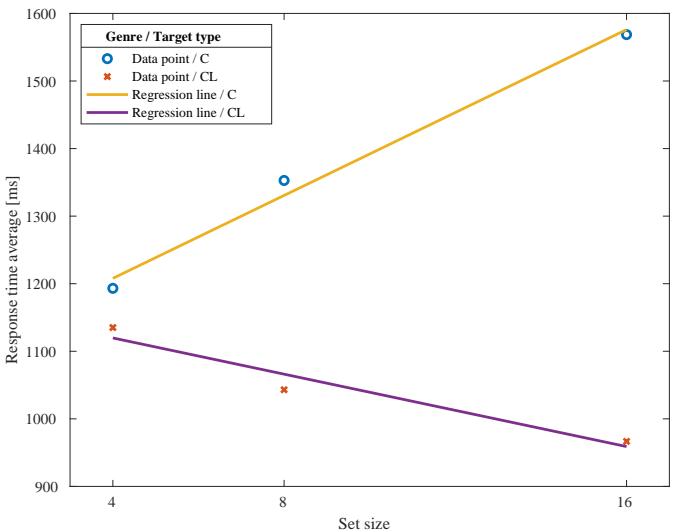
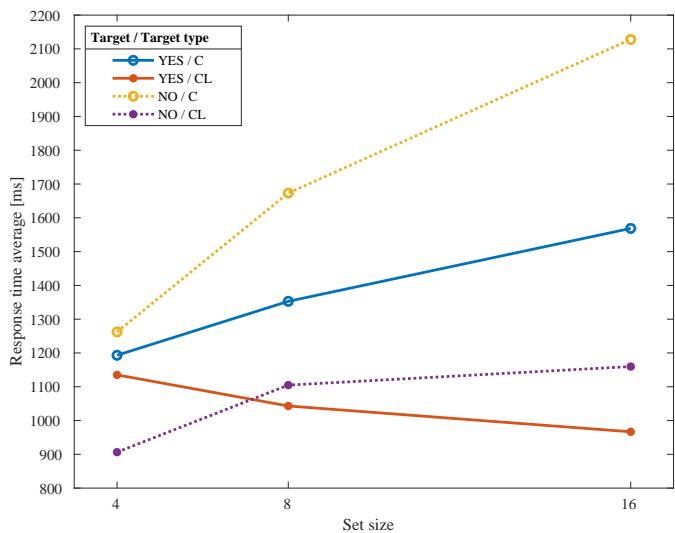
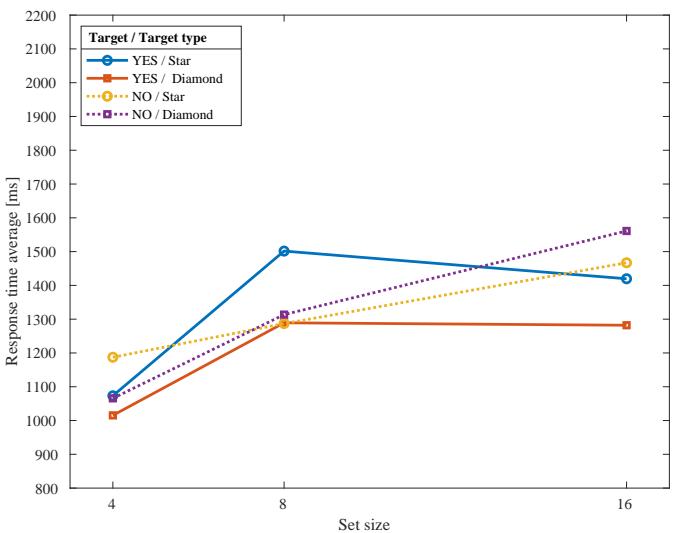


図 3-5: 参加者 A の結果 散布図と回帰直線



(a) 刺激対 1 の刺激数に対する探索時間



(b) 刺激対 2 の刺激数に対する探索時間

図 3-6: <実験 2> 各刺激対に対する探索時間

有意水準 5% で対応のある t 検定を行った結果、刺激対 1 の刺激 C と CL の間で有意な差が認められた。

$$t(13) = 2.9150238 \quad p < 0.05 \quad (3.4)$$

表3-3: 探索時間

参加者番号 (刺激数)	目標刺激: 円						目標刺激: 円+線					
	目標刺激: あり			目標刺激: なし			目標刺激: あり			目標刺激: なし		
	4	8	16	4	8	16	4	8	16	4	8	16
非表示	783.105	872	980.867	857.2	1042.63	1338.65	768.222	761.389	750.778	710.65	754.579	811.5
参加者_A	1193.17	1352.79	1568.67	1262.2	1673.22	2128.1	1135.1	1043.06	966.65	906.4	1104.85	1159.75
非表示	698	979	984.143	791.25	1089.35	1199.75	651.211	621.688	656.412	598.722	594.947	684.737
非表示	888.421	1061.72	1314.4	925.526	1025.88	1095.28	888.421	923.2	1081	845.611	842.947	809.842
非表示	878.778	963.158	1236.47	908.7	1239	1619.55	866.389	747.263	838.471	652.05	744.2	872.7
非表示	849.632	848	994.167	738.105	1079.79	1214	681.75	721.765	703.158	624.611	695	716.947
非表示	755.95	943.105	1037.67	845.053	1053.65	1339.16	745.556	807.389	781.316	677.526	737.25	722.278
非表示	922.5	898.895	1104.83	867.15	1116.9	1424.55	835.8	882	822.278	714.4	790.9	744.526
非表示	913.824	1187.14	1283.6	1049.47	1273.67	1442.74	961.737	802.95	857.1	842.444	790.333	890.05
非表示	776.95	926.526	1023.68	926	1198.58	1756.15	747.263	724.895	738.263	613.053	646.7	755.15
非表示	862.158	966.056	1141.71	890.053	1237	1666.6	821.368	704.5	747.4211	808.2	812.294	831.579
非表示	1000.21	1182	1244.66	1006	1423.25	1679.35	800.8	788.75	826.3	806.26	834.05	941.75
非表示	814.25	990.25	1354.11	1131	1459.3	1841.15	761.55	764.95	840.05	718.15	810.8	1017.6
非表示	909.278	1001.76	1238.6	914	1057.85	1433.95	777.74	721.333	844.579	946.438	931.944	892.667
平均値	874.7304286	1012.314286	1179.112643	936.5505	1212.147857	1512.794286	817.3505	786.7951429	818.1268643	747.4653571	792.1995714	846.5054286
不偏分散	14623.15518	19668.22447	29640.45678	18852.99019	36736.541	80724.11463	14850.098	11007.30814	11615.11971	12556.73028	15278.89779	17008.08359
標準偏差	120.9262386	140.2434727	172.1640403	137.3061914	191.6677881	284.1198948	121.8609782	104.9157192	107.7734648	112.0568172	123.6078387	130.4150436
標準誤差	32.31889671	38.89653383	47.74971348	38.081888565	53.15907984	78.8006807	33.79815425	29.09838502	29.89098103	31.07896925	34.28264619	36.1702513

表3-4: 回帰直線の傾き

参加者番号	目標刺激: 円						目標刺激: 円+線						差
	目標刺激: 円						目標刺激: 円+線						
非表示	16.06991071						-1.435482143						17.50539286
参加者_A	30.67642857						-13.39660714						44.07303571
非表示		20.530625					0.9915714286						19.53905357
非表示	34.93921429						16.5735						18.36571429
非表示	30.43						-0.3654285714						30.79542857
非表示	12.93405357						1.196875						11.73717857
非表示	16.70098214						-2.032321429						18.73330357
非表示	28.13507143						-6.507107143						34.64217857
非表示	19.35846429						-0.4041428571						19.76260714
非表示	21.81151786						2.088696429						19.72282143
非表示	23.10467857						-4.515473214						27.62015179
非表示	45.05892857						6.948214286						38.11071429
非表示	27.75228571						6.975035714						20.77725
非表示	18.57964286						2.491964286						16.08767857
平均値	24.72012883						0.6149496173						24.10517921
不偏分散	75.39627981						48.30711698						88.89262924
標準偏差	8.683103121						6.950332149						9.428288776
標準誤差	2.320656924						1.857554402						2.519816167

3.4 考察

■実験1 結果より、目標刺激がCの場合、目標刺激数に伴って探索時間が増加していることがわかる。それに対して目標刺激がCLの場合、刺激数に対する探索時間はおよそ一定である。これらからt検定を用いて、両刺激(C, CL)には探索非対称性があると判断された。この原因として、目標刺激がCLの探索に対してポップアウト現象が起きていると考えられる。刺激CLは刺激Cに一本の線を足した刺激なので、円刺激に比べて発見しやすい。これに対して、目標刺激がCの探索では、刺激CLにはCの「円」という要素を含んでいるので、ポップアウトは起きず、逐次探索になると考えられる。

■実験2 刺激対1に対する探索時間と、刺激対2に対する探索時間を比較する。刺激対1に比べて、刺激対2は探索時間の偏りは小さい。刺激対2について、StartとDiamondの形状が似ており、判断の時間に差異が生まれないからであろう。刺激対2の目標刺激がない場合は、刺激数に伴って探索時間が増加しており、目標刺激がある場合は、目標刺激に限らず目標刺激数が8から16にかけて一定、または減少している。これは「探索非対称性」があるとは言えない。Star, Diamondの両刺激において、逐次探索でなくポップアウトが生じていると考えられる。

3.5 結論

本実験では、視覚探索課題の心理物理実験をし、目標刺激と妨害刺激が交代しただけで探索時間が大きく異なる「探索非対称性」を検証した。その原因となる「ポップアウト」が起きる刺激の条件と、「逐次探索」が起きる刺激の条件が明らかとなつた。さらに、統計的仮説検定を用いて、刺激Cと刺激CLには探索時間に有意な差が認められ、探索非対称性を認められた。

第4章

脳波測定実験

4.1 実験の目的

脳波は以下のように説明されている。

“神経細胞の間にあるシナプス電位と、後電位などの電極変動の総和を、頭皮上に付けた電極を用いて記録したもの。”
[10]

また、特定の事象に関する脳波を「事象関連電位(ERP)」と呼ぶ。脳波測定では、ERPを電極で測定する。BMI(Brain Machine Interfae)とは、脳からの信号を計測し、それを利用する技術を指す[11]。これまで、BMIを用いた家電の操作や、文字入力操作が研究されている[11]。

■目的 BMIのシステムを体験するために、タイプしたい文字を脳情報から読み取る。具体的に、ターゲット刺激が表示されたときの脳波と、ターゲット刺激でない刺激が表示されたときの脳波を比較する。

4.2 実験の方法と考え方

■実験装置 刺激呈示用装置として、汎用コンピュータと液晶ディスプレイを利用する。脳波計測機器やコンピュータなどの実験装置を表4-1に示す。実験参加者は、学生の20代男性1名である。

表4-1: 実験装置(脳波測定実験)

BMI 装置	電極	ドライ電極
	脳波計測機器	g.USBamp
	分析ソフトウェア	g.BSanalyze
コンピュータ	コンピュータ	Dell Precision M6600 64-bit Operating System
	プロセッサ	Intel(R) Core(TM) i7-2620M SPC @ 2.70GHz 2.70GHz
	メモリ	8GB (7.88GB usable)
	OS	Windows 7 Ultimate Service Pack 1
	MATLAB®	バージョン不明
	Simulink	バージョン不明

用語解説

- **p300**: p300とは、刺激呈示後、潜時約300msに出現する陽性成分のことを指す[12]。
- **参照電極**: 基準電極とも呼ばれる。測定する電位の基準値を測定するための電極である。神経細胞がなく、電位が一定である耳たぶを採用している。
- **接地電極**: 計測した電位を再び戻すために利用する。これにより、データを計測しやすくする。位置は額。

実験の手続き

1. MATLAB®から Simulink のブロックセットを起動し、ドライ電極を頭部に装着する*.
2. 電極 (Fz, Cz, Pz, P4, P07, 0z, P08) を図 4-1 に従ってセットする。参照電極は耳たぶに、接地電極を額にセットする。対象が視覚刺激のため、後頭部に電極が集中している。
3. 実験参加者は、PC 上に表示された 6×6 行列の文字を観察する。灰色の英字または数字が黒の背景上に表示され、その中からランダムな順で 1 文字が 100ms だけ白に変わる。その後、75ms はどの文字も灰色となり、その後また別の文字が白くなる。
4. 実験参加者は集中して画面を注視し、指定された文字が白く変わるたびにその回数を数える。今回は KUT の 3 文字とする。
5. トレーニングモードでは各文字が各行列で 10 回ずつ表示される。
6. トレーニングモードのデータ測定が終了したら、g.BSanalyze を起動し、トレーニングデータを読み込む。KUT が白く表示されたとき (target 条件) とそうでないとき (non-target 条件) の波形規則を学習させ、判定機を作成する。今回は線形判別分析 (LDA) を用いる。
7. target 条件と non-target 条件をそれぞれ加算平均したグラフを出力する。
8. フラッシュ回数は 5 回として、テストモードで同様のタイピング実験をする。

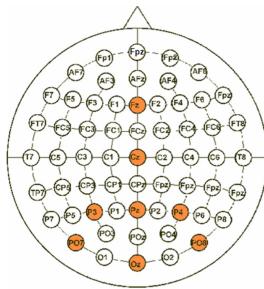


図 4-1: 電極の配置図†



図 4-2: 示される刺激 (例)

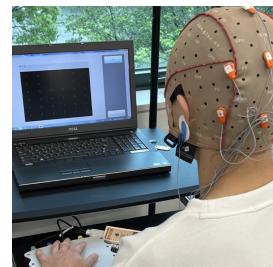


図 4-3: テストモードの実験

4.3 実験の結果

実験結果を図 4-4 に示す。凡例の電極と 2 つのグラフは 1 対 1 で対応している。

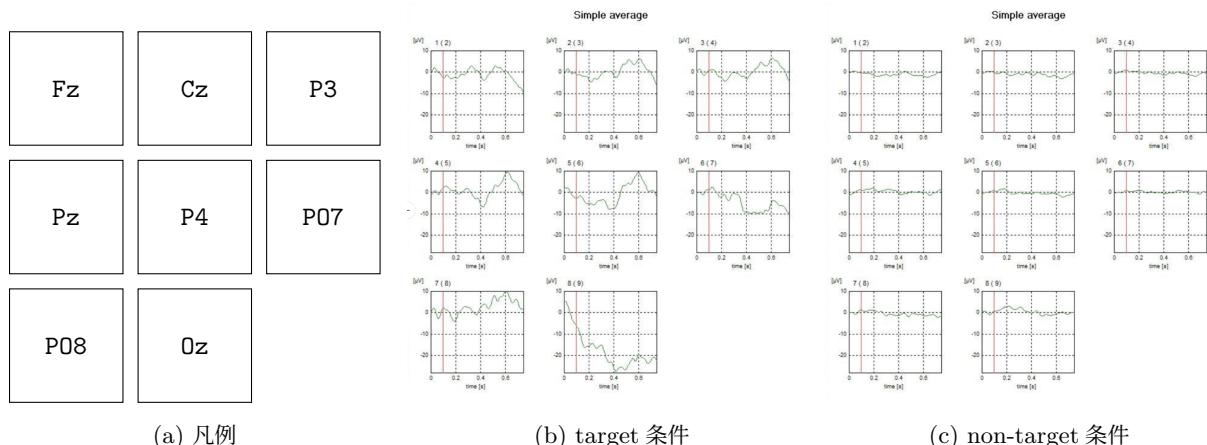


図 4-4: 脳波測定実験実験結果

*サンプリング周波数 256Hz, データに対して 0.1~30Hz のバンドパスフィルタを通し、低周波の大きな電位変化と高周波のノイズを除去する。

†KUT-LMS 課題説明資料 (https://lms.kochi-tech.ac.jp/pluginfile.php/209127/mod_resource/content/12/task0529.pdf) より

4.4 考察

結果（図 4-4）を比較すると、target 条件の方が non-target 条件に比べて電位の振れが大きい。これは、刺激に反応していると考えられる。しかし、特に Oz 電極では電位が負に振れており、この原因は分からなかった。

4.5 結論

本実験により、target 条件下と non-target 条件下では脳波に大きな違いが出た。しかし p300 は刺激呈示後、陽性成分をさしておらず、通常であれば電位が正向きに振れる。結果の一部は大きく負に振れており、この原因は分からなかった。

第 5 章

関連語句

5.1 脳情報デコーディング

脳情報デコーディングとは、脳活動を読み取る技術である。デコーディングとは、符号列を元の状態に戻す作業を指す言葉である。たとえば脳が外界から何らかの刺激を受け取っていた場合、脳情報デコーディング技術を用いて、その刺激について第三者が解読可能である。脳情報デコーディングは、外界の刺激や行動、認知状態と、その時の脳活動との関連性を学習するステップと、その学習結果をもとに脳活動から外界の刺激や行動、認知状態を予測するステップからなる。脳情報デコーディングを用いた研究例として、運動情報の読み取りが挙げられる。ヒト運動野の fMRI 信号を用いてジャンケンを行う例では、ヒトが「チョキ」を出したときの脳活動を計測し、脳活動パターン解析結果により、限りなくリアルタイムに近い状態でロボットに実験参加者と同じジャンケンをさせることができる。今回の課題では、第 4 章の脳波測定実験が脳情報デコーディングにあたる。

[13]

5.2 視線入力インターフェース

視線入力インターフェースとは、視線により、利用者の意思を入力するシステムであり、肢体障がい者のコミュニケーション手段として研究されてきた。視線入力システムには、いくつか方法があるが、ここでは大きく「眼球運動測定法」を取り上げ、眼球運動測定法はさらに、眼電図法 (EOG)、強膜反射法、角膜反射法、瞳孔 - 角膜反射法に分けられている。ここでは角膜反射法について取り上げよう。角膜反射法とは、角膜上に光の反射点を生じさせ、眼球の画像を撮影した後、撮影した眼球の画像から、角膜上の光の反射点と瞳孔を識別する。この結果により眼球の方向を算出する。この技術は、第 1 章の Eye Link II、第 2 章の Tobii Grass 3 に用いられている技術である。

[14]

5.3 顕著性マップ

“顕著性マップとは、人が画像を認識する際に注視しやすい部分を画像で表現したものである。” [16]

第 3 章の図 2-2 で示している Heat Map は、「刺激画像のどこをより注視しているか」を色で表現したものであった。図 2-2 は単なる結果であるが、顕著性マップは画像の物理的特徴を解析して注意が向けられる位置を推定する。大まかな処理を説明する。まず入力画像に対して、6 つの特徴マップ（色、彩度、方位、コントラスト、点滅、運動）を作成する。特徴マップ合成では、6 つの生成した特徴マップを線形和し、顕著性マップを算出する。顕著性マップを用いることで、実験による注視箇所の特定をせずとも、刺激に対する注視箇所がわかるため、テレビ放送などのコマーシャル、包装容姿のデザインなど、商用利用も多く、我々との生活で密にかかわっている。

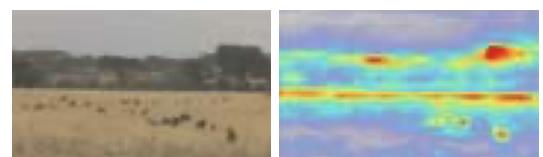


図 5-1: 顕著性マップの例 [15]

[15]

5.4 分散分析

第1章で t 検定を用いて 2 正規母集団の「回帰直線傾きの平均」が異なるか否かを検証した。ここで、3つ以上の正規母集団の母平均が異なるか否かを検証する。このとき、 t 検定を 3 回行う方法では正しい結果を導けない。ここで用いられるのが分散分析 (analysis of variance: ANOVA) である。検定の対象は、母分散ではなく母平均だが、母平均の差を検定するために母分散を利用する事が、この名前の由来である。分散分析には主に、「一元配置分散分析」と「二元配置分散分析」の 2 種類がある。一元配置分散分析は、母平均の差を生む要因が 1 つの場合に用い、二元配置分散分析は、母平均の差を生む要因が 2 つの場合に用いる。

[17, p.170, p.179]

参考文献

- [1] 三浦佳世, 川畠秀明, 横澤一彦. 美感－感と知の統合一（シリーズ統合的認知）. シリーズ統合的認知. 効草書房, 2018.
- [2] 田根健吾, 道又爾. 潜在呈示した情報が選択判断時の視線の動きに与える影響. 基礎心理学研究, Vol. advpub, p. 35.1, 2016.
- [3] 大輔小杉. アイトラッキング技術を用いた地域実践的研究の報告. 静岡文化芸術大学研究紀要, Vol. 17, pp. 177–182, mar 2017.
- [4] 靖久中野. 心理物理測定法. *VISION*, Vol. 7, No. 1, pp. 17–27, 1995.
- [5] 孝恒熊田. 視覚探索. 心理学評論, Vol. 46, No. 3, pp. 426–443, 2003.
- [6] 馬場真哉. Pythonで学ぶあたらしい統計学の教科書. 翔泳社, 2018.
- [7] 武田裕司, 八木昭宏. 視覚探索課題と注意に関する研究動向. 人文論究, Vol. 46, No. 3, pp. 116–130, 12 1996.
- [8] 横澤一彦, 熊田孝恒. 4. 視覚探索. テレビジョン学会誌, Vol. 48, No. 12, pp. 1509–1515, 1994.
- [9] 安藤豊, 大澤秀雄. 応用解析と情報数学（電気・電子・情報系の基礎数学）. 電気・電子・情報系の基礎数学. 東京電機大学出版局, 1993.
- [10] 金策智美. 自己心理学セミナー－自己理解に役立つ13章－. 効草書房, 2018.
- [11] 神作憲司. 脳波による実用的なBMI研究開発. 認知神経科学, Vol. 14, No. 3, pp. 185–192, 2013.
- [12] 加賀佳美, 相原正男. P300基礎. 臨床神経生理学, Vol. 41, No. 2, pp. 80–85, 2013.
- [13] 宮脇陽一, 神谷之康. 《第8回》脳情報デコーディング技術とその応用. 計測と制御, Vol. 50, No. 10, pp. 888–894, 10 2011.
- [14] Dekun Gao. 新たな入力方式と入力画面デザインを考慮した視線入力インターフェースシステムの実現に関する研究. 電気通信大学 学位論文, 2014.
- [15] 上向俊晃, 小峯一晃, 森田寿哉. RJ-002 動画像コンテンツにおける注視点マップと顕著性マップとの関係性に関する考察. 情報科学技術フォーラム講演論文集, Vol. 8, No. 3, pp. 37–42, 08 2009.
- [16] 児玉萌絵, 田中敏幸. 顕著性マップを用いた画質評価法. パーソナルコンピュータ利用技術学会論文誌, Vol. 14, No. 1, pp. 1–7, 2020.
- [17] 本橋永至. Rで学ぶ統計データ分析－マーケティングデータを分析しながら正しい理論と分析力を身につける－. オーム社, 2015.

付録

A 顔画像の印象に関する判断についての眼球運動 (June 1st, 2023)

src. A-1: 1s 前からの眼球運動

```

1 clear; close all;
2 keyData = readmatrix('g0310/exp4i_g0310.csv');
3 tryTime = 20;
4 mzeroC = zeros(500, tryTime);
5 time = (-999:tryTime);
6 for k = 1:tryTime
7     csv = readmatrix(['g0310/g0310.asc_TRIAL_ num2str(k) .csv']);
8     [sizeR, sizeC] = size(csv);
9     ms = size(csv(1,1):2;csv(sizeR,1));
10    ms = ms(1,2); % 欠損値がない場合の理想時間軸
11    expos = zeros(ms,5);
12    expos(:,1) = expos(:,1)*2;csv(sizeR,1);
13    expos(:,1) = expos(:,1)-expos(ms,1);
14    csv(:,1) = csv(:,1)-csv(sizeR,1);
15    expos(:,2:4) = -1; % Defaultで欠損値扱い
16    bin = ismember(expos(:,1),csv(:,1));
17    expos(bin,2) = csv(:,3);
18    expos(bin,5) = csv(:,4);
19    counter = count = 0;
20    for m = 1:ms
21        if(expos(m,2)==1) % 画面状態が1 (表示中)
22            counter = counter + expos(m,5); % x-coordinate 加算
23            count = count + 1;
24        end
25    end
26    x_coordinate_Ave = counter/count;
27    for m = 1:ms
28        if(expos(m,2)==2)
29            if(expos(m,5) <= x_coordinate_Ave)
30                expos(m,3) = 100;
31            elseif(expos(m,5) > x_coordinate_Ave)
32                expos(m,3) = 102;
33            end
34        end
35    end
36    for m = 1:ms
37        if(expos(m,3)==keyData(k,2))
38            expos(m,4) = 1;
39        elseif(expos(m,3)==keyData(k,2) & expos(m,3)~-1)
40            expos(m,4) = 0;
41        end
42    end
43    mzeroC(:,k) = expos(ms-499:ms,4);
44 end
45 ratio = zeros(500,1);
46 for k=1:500
47     ts = mzeroC(k,:);
48     ratio(k) = sum(ts==1)/(tryTime-sum(ts==1));
49 end
50 fig1 = figure;
51 box on;
52 plot(time,ratio);
53 xlabel('time[ms]');
54 ylabel('ratio');
55 exportgraphics(fig1,'../Figures/12_01_graph.pdf','ContentType','vector');

```

src. A-2: 1.2s 前からの眼球運動と近似多項式曲線

```

1 clear; close all;
2 keyData = readmatrix('g0310/exp4i_g0310.csv');
3 tryTime = 20;
4 mzeroC = zeros(600, tryTime);
5 time = (-999:tryTime);
6 for k = 1:tryTime
7     csv = readmatrix(['g0310/g0310.asc_TRIAL_ num2str(k) .csv']);
8     [sizeR, sizeC] = size(csv);
9     ms = size(csv(1,1):2;csv(sizeR,1));
10    ms = ms(1,2); % 欠損値がない場合の理想時間軸
11    expos = zeros(ms,5);
12    expos(:,1) = expos(:,1)-expos(ms,1);
13    csv(:,1) = csv(:,1)-csv(sizeR,1);
14    expos(:,2:4) = -1; % Defaultで欠損値扱い
15    bin = ismember(expos(:,1),csv(:,1));
16    expos(bin,2) = csv(:,3);
17    expos(bin,5) = csv(:,4);
18    counter = count = 0;
19    for m = 1:ms
20        if(expos(m,2)==1) % 画面状態が1 (表示中)
21            counter = counter + expos(m,5); % x-coordinate 加算
22            count = count + 1;
23        end
24    end
25    x_coordinate_Ave = counter/count;
26    for m = 1:ms
27        if(expos(m,2)==2)
28            if(expos(m,5) <= x_coordinate_Ave)
29                expos(m,3) = 100;
30            elseif(expos(m,5) > x_coordinate_Ave)
31                expos(m,3) = 102;
32            end
33        end
34    end
35    for m = 1:ms
36        if(expos(m,3)==keyData(k,2))
37            expos(m,4) = 1;
38        elseif(expos(m,3)==keyData(k,2) & expos(m,3)~-1)
39            expos(m,4) = 0;
40        end
41    end
42    mzeroC(:,k) = expos(ms-599:ms,4);
43 end
44 ratio = zeros(600,1);
45 for k=1:600
46     ts = mzeroC(k,:);
47     ratio(k) = sum(ts==1)/(tryTime-sum(ts==1));
48 end
49 disp(ratio);
50 p = polyfit(time,ratio(:,1),3);
51 fig1 = figure;
52 box on;
53 hold on;
54 plot(time,ratio);
55 x2 = time;
56 y2 = polyval(p,x2);
57 plot(x2,y2,LineWidth=2);
58 hold off;
59 axis([0 1200 0 1]);
60 xlabel('time[ms]');
61 ylabel('ratio');
62 exportgraphics(fig1,'../Figures/12_02_graph.pdf','ContentType','vector');
63 legend('ratio','Approximate polynomial',Location='northwest')

```

B 行動実験と統計的仮設検定 (June 5th and 8th, 2023)

src. B-1: 各刺激に対するグラフ出力

```

1 % import data
2 % 1: 学籍番号
3 % 2: 練習/本番 (1/2)
4 % 3: 試行回数
5 % 4: 目標刺激の有無 (1:あり, 2:なし)
6 % 5: 刺激種類
7 % 6: 刺激経路
8 % 7: 入力状態 (0:正解, 1:失敗, 2:その他のキー, 3:Timeout)
9 % 8: 解答にかかった時間 (ms)
10 fileID = fopen('data1250373.txt', ',', 1);
11 fileID = fopen('data1250373.txt', ',', 1);
12 data = fileread(fileID);
13 practice = data(:,2) == 1; % 練習
14 error = data(:,7) > 0; % Error
15 del = practice | error;
16 data(del,:) = [];
17 % -- 目標刺激 (あり), 目標刺激種類 (マル)
18 target = data(:,4)==1;
19 type = data(:,5)==1;
20 setsize4 = data(:,6)==4; % 4
21 target_type4_size8_mean = mean(data(target & setsize4 & type, 8));
22 target_type4_size16_mean = mean(data(target & setsize4 & type, 8));
23 type = data(:,5)==1;
24 setsize16 = data(:,6)==16; % 16
25 target_type16_size16_mean = mean(data(target & setsize16 & type, 8));
26 % -- 目標刺激 (あり), 目標刺激種類 (マル線)
27 target = data(:,4)==1;
28 type = data(:,5)==2;
29 setsize4 = data(:,6)==4; % 4
30 target_type2_size4_mean = mean(data(target & setsize4 & type, 8));
31 target = data(:,4)==1;
32 type = data(:,5)==2;
33 setsize8 = data(:,6)==8; % 8
34 target_type1_size8_mean = mean(data(target & setsize8 & type, 8));
35 target = data(:,4)==1;
36 type = data(:,5)==2;
37 setsize16 = data(:,6)==16; % 16
38 target_type1_size16_mean = mean(data(target & setsize16 & type, 8));
39 % -- 目標刺激 (あり), 目標刺激種類 (マル線)
40 target = data(:,4)==1;
41 type = data(:,5)==2;
42 setsize8 = data(:,6)==8; % 8
43 target_type1_size8_mean = mean(data(target & setsize8 & type, 8));
44 target = data(:,4)==1;
45 type = data(:,5)==2;
46 setsize16 = data(:,6)==16; % 16
47 target_type1_size16_mean = mean(data(target & setsize16 & type, 8));
48 % -- 目標刺激 (あり), 目標刺激種類 (マル線)
49 target = data(:,4)==1;
50 type = data(:,5)==2;
51 setsize8 = data(:,6)==8; % 8
52 target_type1_size8_mean = mean(data(target & setsize8 & type, 8));
53 target = data(:,4)==1;
54 type = data(:,5)==2;
55 setsize16 = data(:,6)==16; % 16
56 target_type1_size16_mean = mean(data(target & setsize16 & type, 8));
57 % -- 目標刺激 (あり), 目標刺激種類 (マル線)
58 target = data(:,4)==1;
59 type = data(:,5)==2;
60 setsize8 = data(:,6)==8; % 8
61 target_type1_size8_mean = mean(data(target & setsize8 & type, 8));
62 target = data(:,4)==1;
63 type = data(:,5)==2;
64 setsize4 = data(:,6)==4; % 4
65 target_type2_size4_mean = mean(data(target & setsize4 & type, 8));
66 target = data(:,4)==1;
67 type = data(:,5)==2;
68 setsize8 = data(:,6)==8; % 8
69 target_type1_size8_mean = mean(data(target & setsize8 & type, 8));
70 target = data(:,4)==1;
71 type = data(:,5)==2;
72 setsize16 = data(:,6)==16; % 16
73 target_type2_size16_mean = mean(data(target & setsize16 & type, 8));
74 x = [4, 8, 16];
75 mean_type1 = [target_type1_size4_mean target_type1_size8_mean
76 target_type1_size16_mean];
77 mean_type2 = [target_type2_size4_mean target_type2_size8_mean
78 target_type2_size16_mean];
79 mean_type3 = [target_type1_size4_mean ntarger_type1_size8_mean
80 ntarger_type1_size16_mean];
81 mean_type4 = [ntarger_type2_size4_mean ntarger_type2_size8_mean
82 ntarger_type2_size16_mean];

```

```

83 setsize8 = data(:,6)==8; % 8
84 target_type2_size8_mean = mean(data(target & setsize8 & type, 8));
85 target = data(:,4)==1;
86 type = data(:,5)==2;
87 setsize16 = data(:,6)==16; % 16
88 target_type1_size16_mean = mean(data(target & setsize16 & type, 8));
89 target = data(:,4)==1;
90 type = data(:,5)==2;
91 setsize4 = data(:,6)==4; % 4
92 target_type2_size4_mean = mean(data(target & setsize4 & type, 8));
93 target = data(:,4)==1;
94 type = data(:,5)==2;
95 setsize8 = data(:,6)==8; % 8
96 target_type1_size8_mean = mean(data(target & setsize8 & type, 8));
97 target = data(:,4)==1;
98 type = data(:,5)==2;
99 setsize16 = data(:,6)==16; % 16
100 target_type2_size16_mean = mean(data(target & setsize16 & type, 8));
101 x = [4, 8, 16];
102 mean_type1 = [target_type1_size4_mean target_type1_size8_mean
103 target_type1_size16_mean];
104 mean_type2 = [target_type2_size4_mean target_type2_size8_mean
105 target_type2_size16_mean];
106 mean_type3 = [target_type1_size4_mean ntarger_type1_size8_mean
107 ntarger_type1_size16_mean];
108 mean_type4 = [ntarger_type2_size4_mean ntarger_type2_size8_mean
109 ntarger_type2_size16_mean];

```

```

    ntarget_type2_size16_mean];
fig1 = figure;
hold on;
plot(x,mean_type1,'o',LineWidth=2);
plot(x,mean_type2,'*+',LineWidth=2);
plot(x,mean_type3,'o:',LineWidth=2);
plot(x,mean_type4,'*:',LineWidth=2);
hold off;
box on;
lgd = legend({'YES / C','YES / CL','NO / C','NO / CL'},'Location','northwest');
title(lgd,'Target / Target type');
xticks([3 17 900 1600]);
yaxis([800 100 2200]);
axis([3 17 800 2200]);
xlabel('Set size');
ylabel('Response time average [ms]');
exportgraphics(fig1,'../Figures/13_01_graph.pdf','ContentType','vector');

```

src. B-2: 平均値と誤差線の出力

```

clear; close all;
d = importdata('data.csv','','3');
data = d.data;
data(:,1) = [];
[people, setnum] = size(data);
ave_data = mean(data); % 各列の期待値
var_data = var(data); % 各列の不偏分散
std_data = std(data); % 各列の標準偏差
err_data = std_data*sqrt(people); % 標準誤差
setsizes = [4 8 16];
fig1 = figure;
hold on;
errorbar(setsizes, ave_data(1:3), err_data(1:3),'-.',LineWidth=1); % YES / C
errorbar(setsizes, ave_data(7:9), err_data(7:9),'-.',LineWidth=1); % YES / CL
errorbar(setsizes, ave_data(4:6), err_data(4:6),'-.',LineWidth=1); % NO / C
errorbar(setsizes, ave_data(10:12), err_data(10:12),'-.',LineWidth=1); % NO / CL
hold off;
axis([3 17 700 1700]);
xticks([4 8 16]);
lgd = legend({'YES / C','YES / CL','NO / C','NO / CL'},'Location','northwest');
title(lgd,'Target / Target type');
box on;
xlabel('Set size');
ylabel('Response time average [ms]');
exportgraphics(fig1,'../Figures/14_01_graph.pdf','ContentType','vector');

```

src. B-3: 回帰直線の出力

```

clear; close all;
d = importdata('data.csv','','3');
data = d.data;
data(:,1) = [];
[people, setnum] = size(data);
setsizes = [4 8 16];
slope = zeros(people, 3);
for k=1:people;
    y = data(k,1:3);
    p_circle = polyfit(x,y,1);
    slope(k,1) = p_circle(1);
    y = data(k,7:9);
    p_circle = polyfit(x,y,1);
    slope(k,2) = p_circle(1);
    slope(k,3) = p_circle(1) - p_circle(2);
end
mynum = 2;
fig1 = figure;
hold on;
y = data(mynum,1:3);
plot(x,y,'o',LineWidth=2);
p_circle = polyfit(x,y,1);
a = p_circle(:,:1);
b = p_circle(:,:,2);
y1 = a + b*x;
y2 = a + b*x;
y = data(mynum,7:9);
plot(x,y,'x',LineWidth=2);
p_circle_bar = polyfit(x,y,1);
a = p_circle_bar(:,:1);
b = p_circle_bar(:,:,2);
y2 = a + b*x;
y1 = y2;
plot(y1,y2,LineWidth=2);
leg = legend('Data point / C','Data point / CL','Regression line / C','Regression line / CL','Location','northwest');
title(leg,'Genre / Target type');
axis([3 17 900 1600]);
xticks([4 8 16]);
box on;
hold off;
xlabel('Set size');
ylabel('Response time average [ms]');
exportgraphics(fig1,'../Figures/14_02_graph.pdf','ContentType','vector');

```

src. B-4: visualsearch (刺激対 1)

```

global ScreenRect MainWindow
% 初期設定
Screen('Preference','SkipSyncTests');
Screen('Preference','ConserveVRAM', 2^24); % bugによるエラー回避のため
KbName('KeyboardNames'); % キー配置の設定
% 実験バラメタの入力
subject = '1250373'; % 被験者番号 (学籍番号を入力)
setsizes = [4 8 16]; % 刺激セットサイズの条件数
nsetSizes = length(setsizes); % セットサイズの条件数
nTrials = 12; % 練習試行数
nTrials += 20; % 繰り返し数
nTrials = nTrials*2+2*nSetSizes; % 本実験試行数
% その他の実験バラメタの設定
Keys = [KbName('f') KbName('j')]; % 反応用キー
maxWait = 5; % 反応時間の制限時間
fixationDuration = 0.5; % 注視点の表示時間 (秒)
% データ出力用ファイルの設定
formatting = '%d %d %d %d %d %d %d\n';
dataFileName = ['data','.',subject,'.txt'];
if exist(dataFileName,'file') == 0
    count = fprintf(dataFile, '%s\n', [subject, practice/real, trial#, target
    ' non-target', target type, setsizes, error, responseTime']);
    %count = fprintf(dataFile, '%s\n', [subject, practice/real, trial#, target
    ' non-target', target type, setsizes, error, responseTime]);
    fclose('all');
end
try
    ListenChar(2); % プログラム実行中のキー押し反応がコマンドウィンドウやエディタに入力されないようにする
    % 亂数の種設定 (clock関数使用)
    rand('state',sum(100*clock));
    % 明るさの設定
    scrnNum = max([Screen('Screens')]);
    white = blackIndex(scrnNum);
    black = BlackIndex(scrnNum);
    gray = round((white+black)/2); % whiteが255, blackが0, grayが128
    % ウィンドウの提示
    windRect = get(scrnNum,'ScreenSize');

```

```

    [MainWindow, ScreenRect] = Screen('OpenWindow', scrnNum, gray);
    hz = FrameRate(MainWindow); % リフレッシュレート
    % カーソルを隠す
    HideCursor;
    % 5 x 5 cells のマトリクスに刺激を提示するための設定
    nColumns = 5;
    nRows = 5;
    objectSize=40; % 刺激サイズ
    objectRect = [0 0 objectSize objectSize]; % 刺激1個の矩形サイズ
    stimulusRange = objectSize*2; % 刺激1個の中心座標が取りうる範囲
    stimulusRect = [0 0 stimulusRange*nColumns stimulusRange*nRows]; % 刺激全体の提示範囲
    fieldRect = CenterRect(stimulusRect, ScreenRect); % 刺激の提示範囲をスクリーンの中央に設定
    columns = ((1:nColumns) - .5) * stimulusRange;
    rows = ((1:nRows) - .5) * stimulusRange;
    [X, Y] = meshgrid(columns, rows); % 各cellの中心座標
    jitter = 40; % cellの中心座標からのバラツキ度合い
    nPositions = nColumns * nRows; % 位置の数
    % オフスクリーンウィンドウの用意
    stimulus = Screen('OpenOffscreenWindow', MainWindow, gray, stimulusRect); % 刺激描画領域
    circle = Screen('OpenOffscreenWindow', MainWindow, gray, objectRect); % 円刺用の領域
    plusline = Screen('OpenOffscreenWindow', MainWindow, gray, objectRect); % 円プラス線刺激用の領域
    fixation = Screen('OpenOffscreenWindow', MainWindow, gray, objectRect); % 十字の注視点の領域
    % 刺激描画
    stroke = 4; % 線の太さ
    margin = 4;
    ovalRect = [margin margin objectSize-margin objectSize-margin]; % 円の描画範囲
    lineRect = [(objectSize-stroke)/2 objectSize-margin+2*stroke (objectSize+stroke)/2 objectSize]; % 線の描画範囲
    Screen('FrameOval', circle, black, ovalRect, stroke); % 円を描画
    Screen('FrameOval', plusline, black, ovalRect, stroke); % 円を描画
    Screen('FillRect', plusline, black, lineRect); % 線を追加
    % 注視点
    hCrossRect = [0 (objectSize - stroke)/2 objectSize (objectSize + stroke)/2];
    vCrossRect = [(objectSize - stroke)/2 0 (objectSize + stroke)/2 objectSize];
    Screen('FillRect', fixation, black, hCrossRect);
    Screen('FillRect', fixation, black, vCrossRect);
    fixationRect = CenterRect(objectSize, ScreenRect);
    % 教示文の表示
    Screen('Flip', MainWindow, gray); % 画面クリア
    Screen('TextSize', MainWindow, 24);
    offset = 65;
    Screen('DrawText', MainWindow, gray, ['Your task is to detect the presence of the odd target in the distractors.'], ScreenRect(3)/2-offset*6, ScreenRect(4)/2-offset*2);
    Screen('DrawText', MainWindow, ['If the target exists, press "F" key with your left hand.'], ScreenRect(3)/2-offset*6, ScreenRect(4)/2-offset);
    Screen('DrawText', MainWindow, ['If not, press "J" key with your right hand'], ScreenRect(3)/2-offset*6, ScreenRect(4)/2-offset);
    Screen('DrawText', MainWindow, ['Please respond as accurately and quickly as possible.'], ScreenRect(3)/2-offset*6, ScreenRect(4)/2-offset);
    Screen('DrawText', MainWindow, ['The number of practice trials is: ', num2str(xTrials)], ScreenRect(3)/2-offset*6, ScreenRect(4)/2-offset*2);
    Screen('DrawText', MainWindow, ['Press any key'], ScreenRect(3)/2-offset, ScreenRect(4)/2-offset*4);
    Screen('Flip', MainWindow);
    % 在待ち
    FlushEvents('keyDown');
    KbWait;
    WaitSecs(1);
    % 実験開始
    for block = 1:2
        if block == 1
            blockMessage = ['Practice'];
            nTrials = pTrials;
        else
            blockMessage = ['Test'];
            nTrials = xTrials;
        end
        Screen(MainWindow,'FillRect',gray); % 画面クリア
        % キー押して開始
        nTrialsString = num2str(nTrials);
        message = double(['Press any Key to start (' , blockMessage, ' = ', nTrialsString, ')']);
        Screen('DrawText', MainWindow, message, ScreenRect(3)/2-offset*3, ScreenRect(4)/2-10);
        Screen('Flip', MainWindow);
        FlushEvents('keyDown');
        KbWait;
        WaitSecs(0.5);
        trialsequence=randperm(nTrials); % 刺激の提示順をシャッフル
        for trial = 1:nTrials
            Screen(MainWindow, 'FillRect', gray); % 画面クリア
            Screen(stimulus, 'FillRect', gray); % オフスクリーンクリア
            % 刺激条件の設定
            if block == 1 % 練習試行の場合はランダム
                target = cell(2*rand, 1); % 1 = ターゲットあり, 2 = ターゲットなし
                whichtarget = cell(2*rand); % 1 = ターゲットが円プラス線, 2 = ターゲットが円, 3 = ターゲットが円
                setSize = setsizes(cell(nSetSizes*rand));
            else
                target = rem(trialsequence(trial),2)+1; % 1 = ターゲットあり, 2 = ターゲットなし
                whichtarget = cell(trialsequence(trial)/(2*nSetSizes*rTrials)); % 1 = ターゲットが円プラス線, 2 = ターゲットが円
                setSize = setsizes(cell((rem(trialsequence(trial),2*nSetSizes*
                rTrials)+1)/(2*rTrials))); % セットサイズの設定
            end
            positionIndex = randperm(nPositions); % 刺激をランダムに配置
            % 刺激を決定
            for i = 1:setSize
                if whichtarget == 1 % ターゲットが円
                    if target == 1 & i == 1 % ターゲットありの場合
                        item = circle;
                    else
                        item = plusline;
                    end
                else % ターゲットが円プラス線
                    if target == 1 & i == 1 % ターゲットあり
                        item = plusline;
                    else
                        item = circle;
                    end
                end
                % cellの中心からランダムにずらす
                dx = round((ceil(jitter*rand) - jitter)/2); % x軸
                dy = round((ceil(jitter*rand) - jitter)/2); % y軸
                itemRect = CenterRectInPoint(objectRect, X(positionIndex(i)) + dx, Y(positionIndex(i)) + dy);
                Screen('DrawTexture', stimulus, item, objectRect, itemRect);
            end
            % 注視点提示
            Screen('DrawTexture', MainWindow, fixation, objectRect, fixationRect);
            Screen('Flip', MainWindow);
            Screen(MainWindow, 'WaitBlanking', ceil(fixationDuration*hz));
            % 刺激提示
            Screen('MakeTexture', MainWindow, stimulus);
            Screen('DrawTexture', MainWindow, stimulus, stimulusRect, fieldRect);
            Screen('Flip', MainWindow);
            % 反応取得
            initTime = GetSecs; % 時間初期化
            response = []; % 反応初期化
            while isempty(response) && (GetSecs - initTime) < maxWait
                [keyIsDown, KbTime, keyCode] = KbCheck; % キー反応取得
                if keyIsDown
                    response = find(keyCode); % 押したキーのキー取得
                    response = response(1);
                    responseTime = KbTime; % キー押しした時間
                end
            end
            if isempty(response) % キー押ししてなかった場合時間切れとして処理
                err = 3;
            end

```

```

196     feedback = 'Time out!';
197   else
198     RT = 0;
199   end
200   responseKey = find(keys == response); % fまたはjを押したか確認
201   if isempty(responseKey) % fまたはjではなかった場合
202     if response == KbName('ESCAPE') % escキーの場合抜ける
203       break;
204     end
205     err = 2;
206     feedback = ['Wrong key! "F" = YES, "J" = NO. '];
207   elseif responseKey == target % 正答の場合
208     err = 0;
209     feedback = double('Correct! ');
210   else % 誤答の場合
211     err = 1;
212     feedback = double('Wrong! ');
213   end
214
215   % 試行ごとにデータ書き込み
216   datafile = fopen(dataFileName, 'a');
217   fprintf(datafile, formatString, subject, block, trial, target,
218   whichtarget, setSize, err, RT);
219   fclose(datafile);
220
221   % フィルタリング
222   Screen(MainWindow, 'FillRect', gray);
223   feedbacks=double([feedback; RT = ', num2str(RT), ' ms']);
224   Screen('DrawText', MainWindow, feedbacks, ScreenRect(3)/2-offset*2,
225   ScreenRect(4)/-10);
226   Screen('Flip', MainWindow);
227   WaitSecs(1);
228
229 end
230
231 % 終了処理
232 dataFile = fopen(dataFileName, 'a');
233 fprintf(dataFile, '\n');
234 fclose(dataFile);
235
236 Screen(MainWindow, 'FillRect', gray);
237 Screen('DrawText', MainWindow, double('All done! Thank you for participating',
238 ), ScreenRect(3)/-150, ScreenRect(4)/-offset);
239 Screen('DrawText', MainWindow, double('Press any key'), ScreenRect(3)/-150,
240 ScreenRect(4)/-offset*30);
241 FlushEvents('keyDown');
242 KbWait;
243 ListenChar(0);
244 clear Screen;
245 catch
246   ListenChar(0);
247   Screen('CloseAll');
248   psychrethrow(psychlasterror);
249 end

```

src. B-5: visualsearch (刺激対 2)

```

1 global ScreenRect MainWindow
2 % 初期設定
3 Screen('Preference', 'SkipSyncTests');
4 Screen('Preference', 'ConserveVRAM', 2^24); % bugによるエラー回避のため
5 KbName('UnifiedKeyNames'); % キー配置の設定
6
7 % 実験パラメタの入力
8 subject = '1250373'; % 被験者番号 (学籍番号を入力)
9 setSize = [4 8 16]; % 刺激セットサイズ
10 nSetSizes = length(setSize); % セットサイズの条件数
11 pTrials = 12; % 練習試行数
12 rTrials = 20; % 繰り返し数
13 xTrials = rTrials*2+2*nSetSizes; % 本実験試行数
14
15 % その他の実験パラメタの設定
16 keys = [KbName('f') KbName('j')]; % 反応用キー
17 maxWait = 5; % 反応時間の制限時間
18 fixationDuration = 0.5; % 注視点の表示時間 (秒)
19
20 % データ出力用ファイルの設定
21 formatString = '%s, %d, %d, %d, %d, %d, %d \n';
22 dataFileName = 'data', subject, '.txt';
23 if exists('data', 'file') == 0
24   dataFile = fopen(dataFileName, 'a');
25   count = fprintf(dataFile, '%s\n', subject, practice/real, trial#, target
26   , nonTarget, target type, setSize, error, responseTime);
27   count = fprintf(dataFile, '%s\n', '被験者番号, 練習/本試行, 試行数, ター
28   ゲットありなし, ターゲットの種類, 刺激セットサイズ, 誤答, 反応時間');
29   fclose('all');
30 end
31 try
32 ListenChar(2); % プログラム実行中のキー押し反応がコマンドウインドウやエディタ
33 % 入力されないようにする
34 % 乱数の種設定 (clock関数使用)
35 rand('state', sum(100*clock));
36 % 明るさの設定
37 scrnNum = max(Screen('Screens'));
38 white = WhiteIndex(scrnNum);
39 black = BlackIndex(scrnNum);
40 gray = round((white+black)/2); % whiteが255, blackが0, grayが128
41
42 % ウィンドウの提示
43 windRect = get(scrnNum, 'ScreenSize');
44 [MainWindow, ScreenRect] = Screen('OpenWindow', scrnNum, gray);
45 hz = FrameRate(MainWindow); % リフレッシュレート
46 % カーソルを隠す
47 HideCursor();
48 % 5 x 5 セル のマトリクスに刺激を提示するための設定
49 % nColumns = 5;
50 % nRows = 5;
51 nRows = 6;
52 objectSize=40; % 刺激サイズ
53 objectRect = [0 0 objectSize objectSize]; % 刺激 1 個の矩形サイズ
54 stimulusRange = objectSize*2; % 刺激 1 個の中心座標が取りうる範囲
55 stimulusRect = [0 0 stimulusRange*ColumnCount stimulusRange*nRows]; % 刺激全体の
56 提示範囲
57 fieldRect = CenterRect(stimulusRect, ScreenRect); % 刺激の提示範囲をスクリーン
58 % 中央に設定
59 columns = ((1:nColumns) - .5) * stimulusRange;
60 rows = ((1:nRows) - .5) * stimulusRange;
61 [X, Y] = meshgrid(columns, rows); % 各セルの中心座標
62 jitter = 40; % cell の中心座標からのバラツキ度合い
63 nPositions = nColumns * nRows; % 位置の数
64 % オフスクリーン用の意
65 stimulus = Screen('OpenOffscreenWindow', MainWindow, gray, stimulusRect); % 刺激描画領域
66 % circle = Screen('OpenOffscreenWindow', MainWindow, gray, objectRect); % 円削
67 % 激用の領域
68 % cplusline = Screen('OpenOffscreenWindow', MainWindow, gray, objectRect); % 円
69 % ブラス線刺激用の領域
70 star = Screen('MakeTexture', MainWindow, imread('star.png'));
71 syuri = Screen('MakeTexture', MainWindow, imread('syuri.png'));
72 fixation = Screen('OpenOffscreenWindow', MainWindow, gray, objectRect); % 十字
73 % の注視点の領域
74
75 % 刺激描画
76 stroke = 4; % 線の太さ
77 margin = 4;
78 ovalRect = [margin margin objectSize margin objectSize-margin]; % 円の描画範囲
79 ovalRect = [(objectSize-stroke)/2 objectSize-margin*2-stroke (objectSize+
80 stroke)/2 objectSize]; % 線の描画範囲
81 Screen('Frameoval', circle, black, ovalRect, stroke); % 円を描画
82 Screen('Frameoval', cplusline, black, ovalRect, stroke); % 円を描画
83 Screen('FillRect', cplusline, black, lineRect); % 線を追加
84
85 % 注視点
86 hCrossRect = [(objectSize - stroke)/2 objectSize (objectSize + stroke)/2];
87 vCrossRect = [(objectSize - stroke)/2 (objectSize + stroke)/2 objectSize];
88 Screen('FrameRect', fixation, black, hCrossRect);
89 Screen('FillRect', fixation, black, hCrossRect);
90 fixationRect = CenterRect(objectRect, ScreenRect);
91
92 % 教示文の表示
93 Screen('FillRect', MainWindow, gray); % 画面クリア

```

```

94
95 Screen('TextSize', MainWindow, 24);
96 offset = 65;
97
98 Screen('DrawText', MainWindow, ['Your task is to detect the presence of the
99 odd target in the distractors.'], ScreenRect(3)/2-offset*6, ScreenRect
100 (4)/2-offset*2);
101 Screen('DrawText', MainWindow, ['If the target exists, press "F" key with
102 your left hand. If the target is not present, press "J" key with your right hand
103 '], ScreenRect(3)/2-offset*6, ScreenRect(4)/2);
104 Screen('DrawText', MainWindow, ['The number of practice trials is: ', num2str(
105 pTrials), ' and the number of experimental trials is: ', num2str(xTrials
106 )], ScreenRect(3)/2-offset*6, ScreenRect(4)/2+offset*2);
107 Screen('DrawText', MainWindow, ['Press any key'], ScreenRect(3)/2-offset,
108 ScreenRect(4)/2+offset*4);
109 Screen('Flip', MainWindow);
110
111 % キー待ち
112 FlushEvents('keyDown');
113 KbWait;
114 WaitSecs(1);
115
116 % 実験開始
117 for block = 1:2
118   if block == 1
119     blockMessage = ['Practice'];
120     nTrials = pTrials;
121   else
122     blockMessage = ['Test'];
123     nTrials = xTrials;
124   end
125   Screen(MainWindow, 'FillRect', gray); % 画面クリア
126
127 nTrialString = num2str(nTrials);
128 message = double(['Press any key to start ', blockMessage, ' = ',
129 nTrialString, ', ', 'trials']);
130 Screen('DrawText', MainWindow, message, ScreenRect(3)/2-offset*3,
131 ScreenRect(4)/2-10);
132 Screen('Flip', MainWindow);
133 FlushEvents('keyDown');
134 KbWait;
135 WaitSecs(0.5);
136
137 trialsequence=randperm(nTrials); % 刺激の提示順をシャッフル
138 for trial = 1:nTrials
139   Screen(MainWindow, 'FillRect', gray); % 画面クリア
140   Screen(stimulus, 'FillRect', gray); % オフスクリーンクリア
141
142 % 刺激条件の設定
143 if block == 1 % 練習試行の場合はランダム
144   target = cell(2*rand); % 1 = ターゲットあり, 2 = ターゲットなし
145   whichtarget = cell(2*rand); % 1 = ターゲットが円, 2 = ターゲットが
146   % 円プラス線
147   setSizes = setsizes(cell(nSetSizes*rand));
148 else
149   target = rem(trialsequence(trial), 2)+1; % 1 = ターゲットあり, 2 =
150   % ターゲットなし
151   whichtarget = cell(trialsequence(trial)/(2*nSetSizes*rTrials));
152   % 1 = ターゲットが円プラス線, 2 = ターゲットが円
153   setSize = setsizes(cell((rem(trialsequence(trial), 2)*nSetSizes*
154   rTrials)+1)/(2*rTrials))); % セットサイズの設定
155 end
156 positionIndex = randperm(positions); % 刺激をランダムに配置
157
158 % 刺激を決定
159 for i = 1:SetSize
160   if whichtarget == 1 % ターゲットが円
161     if target == 1 & i == 1 % ターゲットありの場合
162       item = star;
163     else item = syuri;
164   end
165 else % ターゲットが円プラス線
166   if target == 1 & i == 1 % ターゲットあり
167     item = star;
168   else item = star;
169 end
170
171 % cell の中心からランダムにずらす
172 dx = round(ceil(jitter*rand)) - jitter/2; % x軸
173 dy = round(ceil(jitter*rand)) - jitter/2; % y軸
174 itemRect = CenterRectOnPoint(objectRect, X(positionIndex(i)) + dx,
175 Y(positionIndex(i)) + dy);
176 Screen('DrawTexture', stimulus, item, objectRect, itemRect);
177
178 % 注視点提示
179 Screen('WaitBlanking', MainWindow, fixation, objectRect, fixationRect);
180 Screen('Flip', MainWindow);
181 Screen(Mainwindow, 'WaitBlanking', ceil(fixationDuration*hz));
182
183 % 刺激提示
184 Screen('MakeTexture', MainWindow, stimulus);
185 Screen('DrawTexture', MainWindow, stimulus, stimulusRect, fieldRect);
186 Screen('Flip', MainWindow);
187
188 % 反応取得
189 initTime = GetSecs; % 時間初期化
190 response = []; % 反応初期化
191 while isempty(response) & (GetSecs - initTime) < maxWait
192   [keyIsDown, KbTime, keyCode] = KbCheck; % キー反応取得
193   if keyIsDown
194     response = find(keyCode); % 押したキーのキーコード取得
195     response = response(1);
196     responseTime = KbTime; % キー押しした時間
197   end
198
199 if isempty(response) % キー押ししてなかった場合時間切れとして処理
200   err = 3;
201   feedback = 'Time out!';
202   RT = 0;
203 else
204   responseKey = find(keys == response); % fまたはjを押したか確認
205   RT = round((responseTime - initTime) * 1000); % 反応時間 (ミリ秒)
206   if isempty(responseKey) % fまたはjではなかった場合
207     if response == KbName('ESCAPE') % escキーの場合抜ける
208       break;
209     end
210     err = 2;
211     feedback = ['Wrong key! "F" = YES, "J" = NO. '];
212   elseif responseKey == target % 正答の場合
213     err = 0;
214     feedback = double('Correct! ');
215   else % 誤答の場合
216     err = 1;
217     feedback = double('Wrong! ');
218   end
219
220 % 試行ごとにデータ書き込み
221 datafile = fopen(dataFileName, 'a');
222 fprintf(datafile, formatString, subject, block, trial, target,
223 whichtarget, setSize, err, RT);
224 fclose(datafile);
225
226 % フィードバック
227 Screen(MainWindow, 'FillRect', gray);
228 feedbacks=double([feedback; RT = ', num2str(RT), ' ms']);
229 Screen(MainWindow, 'DrawText', MainWindow, double('All done! Thank you for participating',
230 ), ScreenRect(3)/-150, ScreenRect(4)/-offset);
231 Screen('DrawText', MainWindow, double('Press any key'), ScreenRect(3)/-150,
232 ScreenRect(4)/-offset*30);
233 Screen('Flip', MainWindow);
234 WaitSecs(1);
235
236 % 終了処理
237 dataFile = fopen(dataFileName, 'a');
238 fprintf(dataFile, '\n');
239 fclose(dataFile);
240
241 ListenChar(0);
242 Screen('CloseAll');
243 psychrethrow(psychlasterror);
244 end

```