マウスの Open-field 試験のための行動計量支援ツールの開発

金子立新[1]　渡辺賢太[1]　前田佐嘉志[1]　廣重法道[1]　鶴田直之[1]

概要：薬学研究における Open-field 法を用いたマウスの行動計量を支援するシステムを開発した。システムは、実験中の作業管理を行う部分と実験動画から自動計量を行う部分から成る。実験の作業管理部分は、更に実験の時間管理をする部分と成果物のファイル管理をする部分から成る。自動計量部分は、既にマウスの移動軌跡検出と移動距離、立ち止まり回数を計測できることに加え、今回は立ち上がり行動の回数計測を実装した。本稿では、これらの実装方式について報告すると同時に、薬学研究者に実利用してもらった後の主観評価をアンケート形式で行ったので、その結果も合わせて報告する。

キーワード：知能、認知、物体検出、GUI

Development of a Support System of Mouse Behavior Measurement for Open-Field Experiment

RISSHIN KANEKO†1　KENTA WATANABE†1
SAKASHI MAEDA†1　NORIMICHI HIROSHIGE†1　NAOYUKI TSURUTA†1

Abstract: We have developed a system to support mouse behavior measurement using the open-field method in pharmaceutical research. The system consists of a part that performs time-management and product-file-management during the experiment and a part that performs automatic measurement from the experiment video. In the automatic measurement part, we have already implemented several movement algorithms, and this time, the measurement of the number of times of the rising action.
In this paper, we report on these implementation methods, and at the same time, perform a subjective evaluation in the form of a questionnaire after having a pharmaceutical researcher actually use it.

Keywords: Intelligence, Recognition, Object Detection, GUI

1. はじめに

現代、動物の尊い犠牲を通じて動物実験が行われており、生命現象の理解に大きな役割を 果たしており、医学・医療に応用され、健康と福祉に計り知れない貢献をしている。動物実験は開発された医療・医薬を人間に適用する前の欠くことのできないステップであり、その必要性と重要性はますます増大していると言える [1]。

薬科系の大学においても、薬物の作用を研究する際に動物実験が行われているが、動物や実験装置にかかるコストや、 研究人員の確保の問題を抱えている。具体例として、本研究の共同研究者が所属する福岡市の第一薬科大学では，マウス と Open-field 法を用いた、Caffeine が動物の一般行動に及ぼす影響を測定する実験が行われており、研究人員の不足から、複数のマウスの同時測定や長時間測定が困難であるという問題を抱えている。

そこで、本論文では、Open-field 試験におけるマウスの行動計量を GUI や画像処理を用いて支援する手法を提案する。

本研究における支援対象処理は、大きく二つに分けると、実験動画からの自動計量と、作業管理とになる。

自動計量には、マウスの追跡(2.2 節)と、立ち上がりの検出(4 節)が含まれる。

作業管理には、実験自体を容易にするための作業タイミングの通知や計測内容の記録と、操作を容易にするための全体の GUI 化が含まれる。

2. 研究背景

2.1 本研究で対象とするマウスの観察実験方法

本研究で支援を行うのは、Caffeine による影響の測定を目的とする Open-field 法のマウス実験である。

Open-field 法とはマウスを用いた動物実験で最も有名な手法の一つであり、一定の面積をほぼ等分に分割し、マウスが一定時間に通過した分割数を自発運動量とする方法である。動物の情動が高まった状態での実験であるため、本能的行動も合わせて観察することができる。

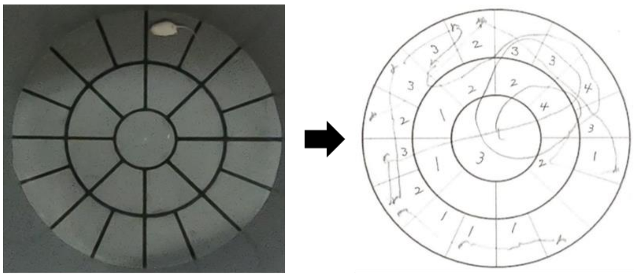
図 2.1(a)は第一薬科大学で使用されている実験装置を真上から撮影したものである。自作の実験装置で、円筒状になっている。従来の手法ではここにマウスを 1 匹ずつ投入して真上から目視で観察し、図 2.1(b)のように手書きで記入していた。また、この測定を Caffeine の投与の直前（pre）、

---

1 福岡大学
†1 Fukuoka University

投与した 15 分後、30 分後、1 時間後、2 時間後、3 時間後の 6 回、それぞれ 4 匹で行うため、計 24 回の測定が必要である。また、間に 3 回の薬物投与も行う。このように作業手順が非常に多く実験者の負担が大きいため、現在は 2 人体制で実験が行われている。

さらにこの実験の後、測定結果を分析し、データ化する作業も負担が大きい。



(a) 実験装置　　(b) 測定の例

図 2.1　Open-field 法実験装置

2.2 自動計量における提案の概要

前節で述べたような、自発運動量の計測のための目視によるマウスの追跡の負荷を軽減するために、まず、画像処理によるマウスの追跡を提案する。

具体的には、OpenCV の Tracking API[2]を利用する。追跡時には動画の先頭フレームからマウスの初期位置を与え，トラッキングを開始する。この機能を含め、マウスの動いた軌跡と距離、停止した位置と時間、時間毎の運動量のグラフを出力する機能を開発した。

次に、本能的行動の目視による発見の負荷を軽減することを考える。今回は、本能的行動のうち、「立ち上がり」の自動検出を提案する。

2.3 作業管理における提案の概要

負荷軽減の次の対象としては、システムの操作性の向上と、Open-field 試験中の作業支援の 2 点を目的とし、ユーザインターフェイスの開発を行った。

Open-field 試験ではマウスの行動の計測や薬物の投与などの多くの手順があるため、作業負担が大きい。そこで、ユーザインターフェイスにタイマーやアラーム機能などを搭載することで、作業負担を軽減させる。また、入力から先行研究で開発されたシステムの操作、出力までのすべてをユーザインターフェイス上で行えるようにすることで操作性を向上させる。言語は Python を使用した。アプリケーション開発者でない薬科大関係者が通常使っている Windows PC 上で、開発言語・環境を意識することなく操作できるようにするために、これを実現可能な GUI アプリ化を行った。GUI ライブラリには wxPython を使用した。

3. システムの概要

2.3 節で述べた理由に基づき、GUI アプリを作成する。

システムの構成図を図 3.1 に示す。図における「システム本体」は自動計量を行うシステム、「GUI」はユーザが直接操作するユーザインターフェイスを表す。「中間成果物」は Open-field 実験時の測定開始時間などの記録すべき事項を Excel ファイルに保存したものである。このシステムにより、GUI の操作のみでシステム全体を操作することができるようになる。

現時点では、システム本体には、自動計量のうち、マウスの追跡機能の追加のみ完了している。別プログラムとしては開発済みのマウスの立ち上がり検出機能、および今後開発される未実装のマウスの他の本能的行動検出機能は、今後システム本体に追加していき、ユーザインターフェイスで操作できるように反映させていく。

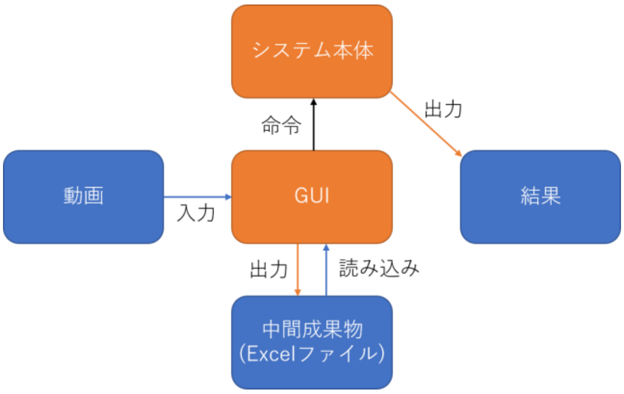


図 3.1　システム構成図

4. 立ち上がりの自動検出における提案手法

前述したように、実験環境は壁に囲まれた円内でのマウスの行動を真上から観察する。この実験環境では、壁によじ登ろうとする場合に限定すれば、より平易で検出精度の高い方法を採用可能である。しかし、精度が低くなるとしても、壁以外の部分でも検出を行いたい。従って、マウスの立ち上がりの検出方法として、①壁によじ登ろうとするときの立ち上がりの検出方法、②円内での立ち上がりの検出方法の 2 種類を併用することを提案する。以下にそれぞれの検出手法を述べる。

4.1 壁によじ登ろうとするときの立ち上がりの検出方法

図 3.1 は、壁によじ登ろうとしたときのマウスの出力画像である。マウスを囲んでいる赤い矩形領域（bounding box）は最初に実現した機能であり、逐次追跡しているマウ

スの位置を表している。ここで、マウスは円の中を行動していることに着目したとき、マウスが壁で立ち上がりしたとき矩形領域が円の外から出ていることが観察できた。以下が壁によじ登ろうとするときの立ち上がりの検出の流れである。

(1) ネズミの行動範囲は円上なので、動画内での円の中心点 p0、半径 r を求める。
(2) 矩形領域の中心点の座標 p3 を求める。
(3) 矩形領域の中心点が円の外側に出たとき立ち上がり行動をしたとみなす。つまり、p0 と p3 の距離が半径 r よりも大きいときである（図 3.2)。

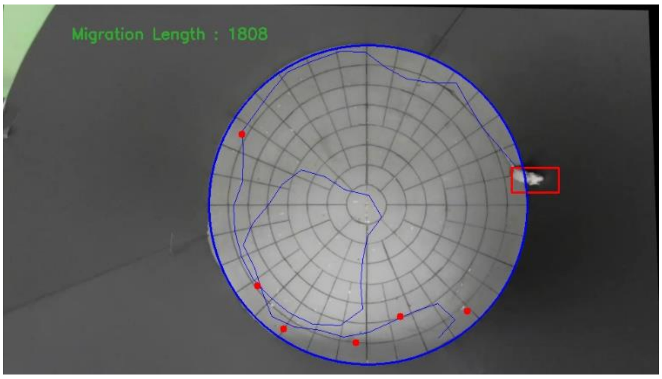


図 4.1　壁での立ち上がりをしたときのマウスの様子

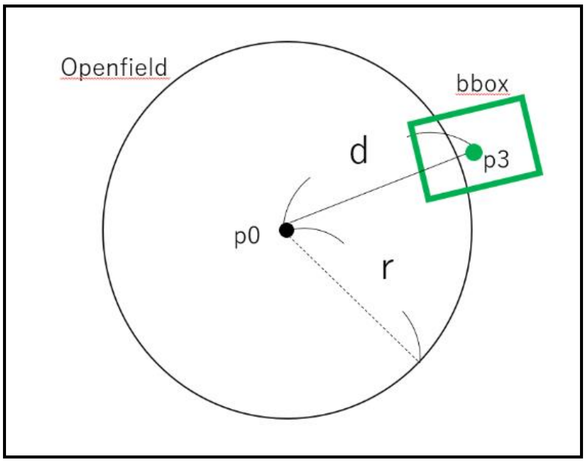


図 4.2　壁によじ登ろうとするときの立ち上がりの検出方法

4.2 円内での立ち上がりの検出方法

4.2.1 検出の評価方法の決定

マウスを真上から観察したとき、歩き回っている場合と立ち上がりをした場合ではマウスの大きさ、形に違いが見られる。立ち上がりを真上から観察したときマウス面積は歩き回りと比べ小さくなる傾向にある。またマウスの輪郭をとらえたとき、立ち上がりをしているマウスの方がより円形に近い形をしていることが観察できた。このことから円内でのマウスの立ち上がりでの検出を

(1) マウスの面積
(2) マウスの円形度

のそれぞれに対して分析し、いずれかを検出の判定に用いる。

分析方法は、面積または円形度の値に対する立ち上がりと歩き回りの個数のヒストグラムを求め、立ち上がりと歩き回りを識別できるような面積または円形度の閾値が存在しているかを確認することにより行う。

分析の結果、今回は、面積を用いることにした。

4.2.2 検出の手順

検出の具体的な手順(図 4.3)は次の様になる。

（手順１）入力画像からフレームごとに二値化を行い、マウスの面積または円形度を計算する。

（手順２）面積または円形度を閾値とし立ち上がりのみを検出させる。

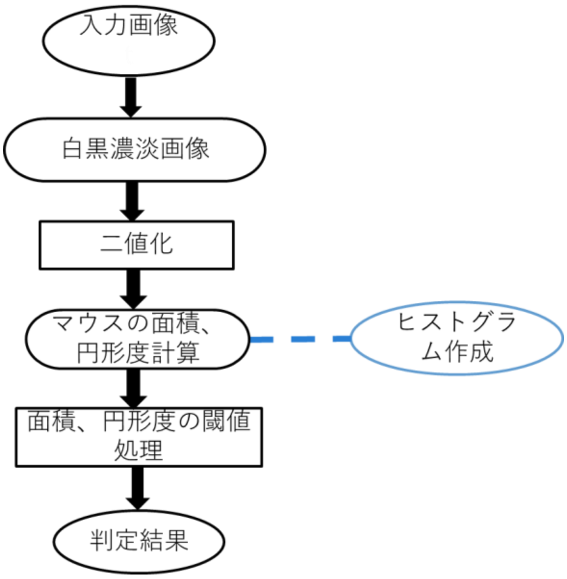


図 4.3　円内での立ち上がりの検出手順

4.3 適応的二値化処理

前述した二値化処理について、今回使用する技術的な説明を行う。単純な二値化処理ではある画像に対して一つの閾値を与えて閾値処理をし、画素値が閾値より大きければある値 255(白色)を割り当て，そうでなければ別の値 0(黒色)を割り当てる。

しかし撮影条件により画像上の座標ごとに光源環境の影響が異なるような状況に対しては期待する結果が得られない。そういう状況では「適応的二値化処理」を使うと有効である。適応的二値化処理では，画像の小領域ごとに閾値の値を計算する。そのため領域によって光源環境が変わるような画像に対しては，単純な閾値処理より良い結果が得られる[3]。

4.4 ラベリング処理

二値化画像処理された画像において、白の部分（または黒の部分）が連続した画素に同じ番号を割り振る処理をラベリングと言う。

OpenCV の connectedComponentsWithStats は、2 値画像中のオブジェクトを検出するメソッドである。その戻り値には、オブジェクトの位置、オブジェクトを囲む矩形情報、オブジェクトの面積（ピクセル）や重心などの情報も含まれている。このオブジェクトの位置とオブジェクトの面積を利用することで、小さすぎるオブジェクトあるいは大きすぎるオブジェクトを取り除くことができる[4]。このメソッドを用いることでマウスの領域のみを検出させることを試みる。

5. ユーザインターフェイスの開発における提案手法

5.1 要件定義

第一薬科大学側の要望を聴取した結果、必要であると考えられる機能を以下に示す。

- 自由記入欄（実験内容識別、メモ）
- ストップウォッチ
- タイマー
- アラーム
- 時刻記録機能
- Excel ファイルへの保存
- Excel ファイルを開く
- 動画ファイル選択
- ボタン押下による実行

5.2 開発システム

前述の要求定義に従って開発したユーザインターフェイス画面を図 5.1 と図 5.2 に示す。

図 5.1 は実験前と実験中に使用する実際のユーザインターフェイスの画面である。図 5.2 は実験後のデータ化する際に使用する画面である。この 2 つの画面は画面上部のタブによって切り替えることができる。

日付 2020/01/13

図 5.1 ユーザインターフェイス画面(1)

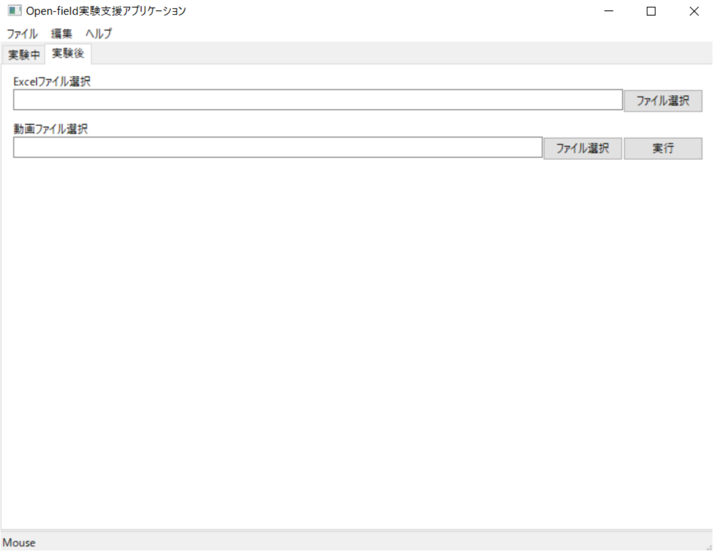


図 5.2 ユーザインターフェイス画面(2)

6. 評価実験

6.1 壁によじ登ろうとするときの立ち上がり

4.1 節の検出方法を元にしたプログラムを提案システムに追加し、実行した結果の画像が図 6.1 である。図 6.1 において、緑色の点が立ち上がりを検出した位置である。

今回は３つの動画で検証を行い、立ち上がりの検出の精度を混同行列で表現した（表 1)。今回は立ち上がりがなく検出がないと判定するフレーム画像は非常に多く存在することから斜線を引いている。したがって正解率による精度の評価は行わず、再現率と適合率から求めることができる F 値により検出精度を評価した。表 2 より壁によじ登ろうとする立ち上がりは F 値 81.63%という精度で検出させることできた。

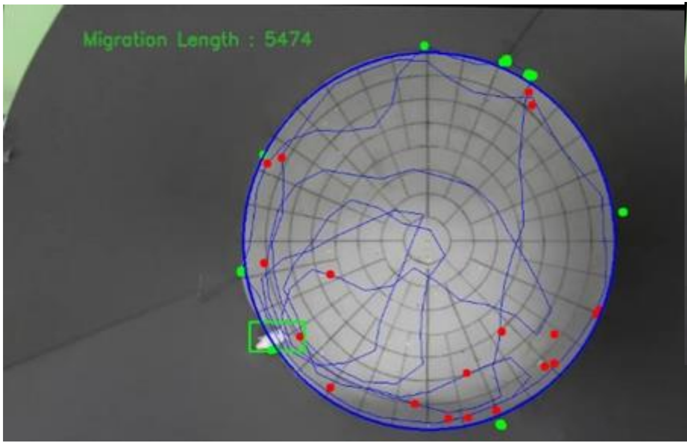


図 6.1 壁によじ登ろうとするときの立ち上がりの位置の描画（緑色の点）

表 1 性能評価表

	検出あり	検出なし
立ち上がりあり	20	6
立ち上がりなし	3	

表 2 壁によじ登ろうとする立ち上がりの性能評価

再現率	79.92%
適合率	86.96%
F値	81.63%

6.2 円内での立ち上がり

マウスの面積が 200～300px の場合は立ち上がりと判定し、その時の位置を緑色の点で描画したときの出力画像が図 6.2 である。今回も３つの動画を使用し混同行列での性能評価を行った (表 3)。表 4 から円内での立ち上がりは F 値 75.78％の精度で検出させることができた。

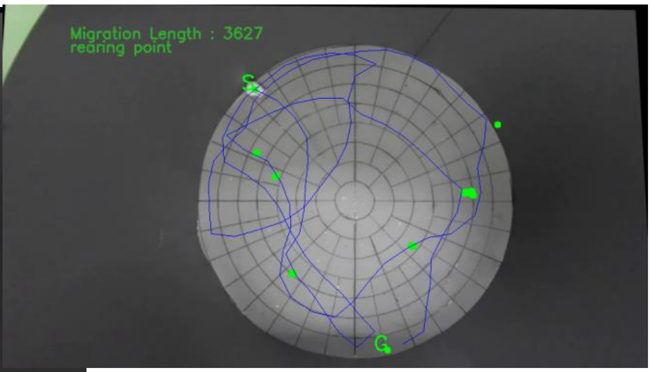


図 6.2 円内での立ち上がり位置の出力画像

表 3 性能評価表

	検出あり	検出なし
立ち上がりあり	14	7
立ち上がりなし	2	

表 4 円内での立ち上がりによる性能評価

再現率	66.76%
適合率	87.50%
F値	75.78%

6.3 ユーザインターフェイスのアンケート評価

以下の 2 点の項目において、本研究の共同研究者である第一薬科大学の小野信文教授と小野教授の研究室の学生 1 名に 5 段階評価を頂いた。

- ユーザインターフェイスによって作業効率は向上したか
- ユーザインターフェイスの使いやすさ

各項目についての評価を表 5 に示す。

また、全体的な評価として以下の意見を頂いた。

- タイマー機能によって他の作業を同時に行うことができるため、作業効率が上がり、負担が減った。
- 撮影開始時刻が表示されるため、現在行っている作業が容易に把握でき、マウスの取り違えが起きにくくなった。
- 実験内容の変更や現在の実験から派生した実験、全く別の実験などを行う場合にも対応できるように、表の項目の追加、削除などをユーザが自由に変更できるようになると良い。

表 5　アンケート評価結果

項目	評価	
	教授	学生
ユーザインターフェイスによって作業効率は向上したか	3	5
ユーザインターフェイスの使いやすさ	4	4

7.　結論

本研究では、Open-field 法を用いたマウスの行動計量支援ツールの開発を行った。

まず、立ち上がりの検出において、２種類の手法を提案した。壁によじ登ろうとする立ち上がりの検出は 3 つの動画を用いて、F 値 81.63％で検出することができた。円内での立ち上がりの検出は 3 つの動画を用いて、F 値 75.78％で検出することができた。

次に、ユーザインターフェイスの開発を提案した。

研究の目的であるシステムの操作性の向上については、動画のフレームサイズとパスの自動取得機能と画像処理の実行をボタン押下のみで行えるようにすることで解決を図り、ユーザインターフェイスの使いやすさにおいて高評価を得たため、解決できたと考える。

同じく目的である実験中の作業支援については、タイマーやアラーム機能などによって解決を図り、ユーザインターフェイスによって作業効率は向上したかにおいて、主にシステムの操作を行って頂いた第一薬科大学の学生から高評価を得たため、解決できたと考える。

今後の改良としては、立ち上がり検出プログラムの、提案システム本体への統合や、他の本能的行動の検出が考えられる。

謝辞　共同研究者である第一薬科大学の小野信文教授、その他ご協力頂いた関係者の皆様に、謹んで感謝の意を表する。

参考文献

[1]“動物実験について“, 日本生理学会, http://physiology.jp/guidance/4804/Word
[2]Satya Mallick. Object tracking using opencv (c++/python) — learn opencv., https://www.learnopencv.com/object-tracking-using-opencv-cpp-python/
[3]画像の閾値処理, http://lang.sist.chukyo-u.ac.jp/classes/OpenCV/py_tutorials/py_imgproc/py_thresholding/py_thresholding.html#thresholding.
[4]OpenCV を使用して画像中のオブジェクトの輪郭を検出する方法, https://axa.biopapyrus.jp/ia/opencv/オブジェクト検出.html.