

# スマートフォン用ウェブページとアプリにおける タップ成功率推定ツール Tappy の実用化

山中 祥太<sup>1</sup> 薄羽 大樹<sup>1</sup> 山下 郁矢<sup>1</sup> 笹谷 奈翁美<sup>1</sup> 佐藤 潤一<sup>1,†1</sup> 山口 修司<sup>1</sup>

**概要:** スマートフォンでウェブサイトやアプリを利用するとき、ボタンやリンクなどのユーザインタフェース (UI) 要素をタップ操作で選択することが一般的である。HCI 分野ではこれまでに様々なタップ操作の成功率推定モデルが提案され、それらは UI デザインに活用できると主張されることが多かったが、実際の開発現場で利用された例は見当たらない。我々はそのようなモデル活用の実例として、Dual Gaussian Distribution Model を用いた 2 つのツール群「Tappy」を一般公開した。1 つはウェブアプリケーション版であり、URL を指定するだけでウェブページ上のタップ可能な要素を特定し、その大きさに基づいてタップ成功率を推定する。もう 1 つは Figma プラグイン版であり、設計中のアプリなどに含まれる UI 要素のタップ成功率を表示する。本稿では Tappy の理論的背景および機能を説明し、公開後の利用状況およびウェブアプリ版の社内利用者インタビュー結果について述べる。

## 1. はじめに

スマートフォン向けのアプリやウェブサイトを開発するにあたって、ユーザが快適に利用できるユーザインタフェース (UI) を設計することは重要である。ユーザの視点では、操作ミスが頻発する UI にはネガティブな印象をいただき、また開発者の観点では、そういった UI はユーザの離脱につながるため、可能なかぎり避けるべきである。スマートフォン上での基本的な操作の 1 つが、ボタン・ハイパーリンク・リンク付き画像などの UI 要素 (ターゲットと呼ぶ) をタップすることなので、UI デザインに関する多くの書籍が、ターゲットを簡単にタップできるように大きく設計することを推奨している [1], [2], [3], [4].

ターゲットの適切な大きさについては、各種のガイドラインを参照しつつアプリやウェブページを設計するのが一般的である。たとえば Android のデザインガイドラインは、9 mm 以上のターゲットサイズを推奨している [5]。しかし、画面上に多くの UI 要素を表示しなければならず、各ターゲットのサイズが制限される場合もある。このとき、各 UI 要素が何%の確率でタップできるかは、ターゲットサイズに基づいて推定可能である [6], [7] ため、デザイナーがそのつどタップ成功率を計算できれば理想的である。

また、前述のように UI 設計のためのガイドラインやタップ成功率推定モデルが存在するが、既存の UI において見

習うべき点や改善点を分析することも重要である。たとえば、あるウェブページの各 UI 要素のタップ成功率を推定できれば、他社のウェブサービスのユーザビリティを分析したり、自社の既存サービスの改善点を発見することに役立つ。しかし、他社のウェブページの完全なソースコード (サーバ側のコードなど) は入手不可能であり、また自社サイトであっても他部署からソースコードを入手できない場合があるため、ソースコードの入手可能性によらずに、ウェブページ上のタップ可能な UI 要素を分析するためのツールがあると便利である。

そこで我々は、ソースコードの入手可能性によらずに、ウェブページ上のタップ可能な UI 要素を分析するためのウェブアプリ「Tappy」を 2023 年に自社内で公開した [8], [9]。さらに、スマートフォン用アプリやウェブページを設計するためのソフトウェア Figma [10] 向けに、実装中の UI 要素のタップ成功率を表示するプラグインを 2022 年に自社内で公開した [11]。これらはいずれも、現在は社外ユーザが利用可能な形態で公開済みである<sup>\*1\*</sup><sup>\*2</sup>。ウェブアプリ版は、既に完成されて公開済みの UI を分析するのに対して、Figma プラグイン版はこれから公開される UI をより良くするためのものであり、相補的な役割を果たす。本稿では Tappy を実装するにあたっての理論的背景を説明したうえで、上記 2 つのツールの機能および利用状況について述べ、さらにプロのデザイナーやエンジニアの社員がウェブアプリ

<sup>1</sup> LINE ヤフー株式会社

<sup>†1</sup> 現在, SB Intuitions 株式会社

<sup>\*1</sup> <https://tappy.yahoo.co.jp/>

<sup>\*2</sup> <https://www.figma.com/community/plugin/1425006564066437139/tappy>

り版 Tappy を活用した事例について報告する。

## 2. 関連研究

### 2.1 ウェブページやアプリの UI 分析手法

Prazina ら [12] が詳細にサーベイしているように、ウェブページの分析目的として多いのは広告の挿入位置の最適化 [13] や、ドキュメントオブジェクトモデル (DOM) の分析によるフィッシング攻撃検出などである [14]。DOM 分析を UI 操作に活用した例として、Bubble Cursor [15] を Chrome ブラウザの拡張機能に実装したものがある [16]。しかしこの実装を見ると、HTML タグの名前と属性によってクリック可能領域を検出しており [17]、クリックイベントが登録されている要素を全ては特定できない。

Prefab は、ウィジェットの視覚的特徴を検出してクリック可能領域を特定するツールキットである [18], [19], [20]。「OK/Cancel」ダイアログや、ウィンドウの最小化/最大化/閉じるボタンなどが検出可能だが、各 OS における標準的な視覚的外観を持っている必要がある。Prefab の改良版にはクリック可能領域を手動で指定する機能があるが、アドホックで手間がかかるのがリミテーションである [21]。

スマートフォン画面上の視覚的情報に基づいて UI 要素を特定する技術も盛んに研究されている。アプリなどのスクリーンショットから、コンピュータビジョン (CV) 技術によって画像やテキストなどの要素を特定する方法 [22] や、スクリーンショットとビュー階層を分析して UI 要素を自動検出する方法 [23]、スクリーンショット上の要素がタップ可能かを深層学習で推定する方法 [24], [25], [26], [27] などがある。画面を自動的に操作してタップ可能な要素を特定するツール [28] や、アイコンなどに紐づけられている機能を推定するシステムなども提案されている [29]。

以上の先行研究をまとめると、ウェブページの DOM やアプリの外観を分析して、クリック・タップ可能な領域を自動的に識別する方法が多く研究されているが、CV 技術などに由来する検出精度の上限や、手間の問題がリミテーションとされている。これに対してウェブアプリ版 Tappy では、あるテキストにリンクが設定されているか、さらにはテキストの周囲領域までタップ可能に設定されているかなど、外観から判断不能な場合にも対応している。また、上記のような分析ツールが実際にプロのエンジニアやデザイナーの業務に役立つのかを追跡調査した文献がなかったのに対し、本稿ではインタビュー調査によって有用性のエビデンスを得ている。Figma プラグイン版 Tappy では、Figma 上で選択されている UI 要素のサイズが取得できるため、タップ可能領域特定に関する問題は生じない。

### 2.2 ポインティングの成功率推定モデル

近年の HCI 分野では、フィッツの法則 [30] に代表される操作時間推定モデルに加えて、クリック・タップ座

標の分布や成功率を推定するモデルの研究が盛んである [6], [31], [32], [33], [34], [35]。その中でも Bi らの Dual Gaussian Distribution Model [6], [36] が強く支持されており、タスク条件を変えて派生形のモデルを導出する研究が多くある。Tappy もこのモデルによってタップ成功率を推定しており、本節でその仕組みを説明する。

スタイラスなどのピンポイントに目標位置を狙えるデバイスを用いると、タップ座標が正規分布に従い、ターゲットサイズ  $W$  に比例してタップ座標の標準偏差  $\sigma$  も大きくなることが知られている [37], [38], [39]。よってタップ座標の分散  $\sigma^2$  について、比例定数  $\alpha$  を用いて次式を得る。

$$\sigma^2 = \alpha W^2 \quad (1)$$

これは非常に小さなターゲットであっても、時間をかければ分散がゼロに近くなる（毎回ほぼ同じ座標をタップできる）ことを示している。しかし指でのタップ操作では、時間をかけても小さいターゲットに対してはタップ座標が試行ごとに変動する。Bi らはこれをタップ精度の下限  $\sigma_a$  とみなし、最終的に観測されるタップ座標の標準偏差  $\sigma$  に対して、ターゲットサイズ  $W$  とは独立に影響すると考えた：

$$\sigma^2 = \alpha W^2 + \sigma_a^2 \quad (2)$$

Bi らの実験では、スマートフォン上の円形ターゲット（直径  $W$  が 5 種類）を繰り返しタップさせ、観測された  $\sigma^2$  と実験で用いた  $W^2$  を回帰分析して定数  $\alpha$  と  $\sigma_a^2$  を算出した [40]。これによって、未検証のターゲットサイズ  $W$  におけるタップ座標の標準偏差  $\sigma$  を推定できる。なお、 $x$  軸・ $y$  軸それぞれのタップ座標が正規分布に従い、回帰式 (2) が適合することが他の実験でも報告されている [31], [41], [42]。

Dual Gaussian Distribution Model によれば、タップ座標は  $x$  軸・ $y$  軸について独立だと仮定でき（相関  $\rho$  がゼロ）、このときのタップ座標の確率密度関数は、

$$P(x, y) = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left(-\frac{x^2}{2\sigma_x^2} - \frac{y^2}{2\sigma_y^2}\right) \quad (3)$$

である。タップ座標がターゲットの範囲  $D$  に入る確率は、

$$P(D) = \iint_D \frac{1}{2\pi\sigma_x\sigma_y} \exp\left(-\frac{x^2}{2\sigma_x^2} - \frac{y^2}{2\sigma_y^2}\right) dx dy \quad (4)$$

である。画面上の長方形ターゲット（横幅  $W$ 、縦幅  $H$ ）の範囲を  $D = \{(x, y) | -\frac{W}{2} \leq x \leq \frac{W}{2}, -\frac{H}{2} \leq y \leq \frac{H}{2}\}$  と定義すれば、 $x$  軸・ $y$  軸の積分を独立に計算できるので、

$$P(D) = \int_{-\frac{W}{2}}^{\frac{W}{2}} \frac{1}{\sqrt{2\pi}\sigma_x} \exp\left(-\frac{x^2}{2\sigma_x^2}\right) dx \int_{-\frac{H}{2}}^{\frac{H}{2}} \frac{1}{\sqrt{2\pi}\sigma_y} \exp\left(-\frac{y^2}{2\sigma_y^2}\right) dy \quad (5)$$

となる。積分計算の箇所をガウスの誤差関数  $\text{erf}(\cdot)$  に置き換えれば、長方形ターゲットをタップする成功率  $SR$  は、

$$SR = \text{erf}\left(\frac{W}{2\sqrt{2}\sigma_x}\right) \text{erf}\left(\frac{H}{2\sqrt{2}\sigma_y}\right) \quad (6)$$

となる。ここで  $\sigma_x$  と  $\sigma_y$  の値を、ターゲットサイズ  $W$  と  $H$  に基づいて式 (2) から推定すれば、未検証のターゲットサイズに対するタップ成功率も予測できる。

Bi らのモデルは当初、静止したターゲットを指でタップするタスクを対象としていた。その派生形として、静止中や移動中のターゲットをマウスで選択するタスク [33], [43], [44], バーチャルリアリティ空間で球体ターゲットを選択するタスク [45], 任意形状のターゲットをスタイラスで選択するタスク [32], 長方形ターゲットに斜め方向から進入してタップするタスク [34] などがあり、いずれも Dual Gaussian Distribution Model の修正モデルで成功率を推定している。

これらのモデルは UI 設計に有用であると説明されることが一般的であり、前段落に挙げた全ての論文がそういった主張を含んでいる。たとえば Ma らは「UI デザイナーは我々の提案モデルによってタッチスクリーンの UI を設計・評価できる」(it can be used by UI designers in designing and evaluating touchscreen user interfaces) と述べている [34]。しかし我々が調査した限りでは、HCI 分野で提案されたこれらのモデルを活用して、プロのデザイナーが開発現場で UI を設計・評価したことを報告した論文は見当たらず、これが先行研究のリミテーションであると考えた。HCI 分野は応用科学だが産業会に対するインパクトが欠如しているという批判とそれへの反論があるが [46]、この問題は UI 操作のモデル化研究にも当てはまると我々は考える。導出したモデルをデザイナーの業務に役立ててもらうには、モデルを使いやすい形態で提供することが必要であると考え、我々は本稿で説明する 2 つのツールを開発・公開した。

### 3. ウェブアプリ版 Tappy

ウェブアプリ版では、任意の URL を入力することで、それをスマートフォンで閲覧したときの画面に表示される UI 要素が自動的に分析される。以下では使用方法と実装の詳細、および一般公開してからの利用状況について述べる。

#### 3.1 使用方法

Tappy のウェブサイト\*<sup>1</sup>にアクセスしたユーザは、分析対象ページの URL を入力し、以降で述べるオプションを設定して「分析する」ボタンを押す。すると Tappy は、指定されたページをグレースケール化したスクリーンショットを表示し、タップ可能な要素を外接矩形で囲む (図 1)。その中から 1 つの要素をクリックすると、タップ成功率、横と縦のサイズ (ピクセル単位と mm 単位) が表示される。クリックしたカーソル位置に 2 つ以上の要素が重なっているときは、全ての候補 (図 1 右の「#1」と「#2」のラベルが付いた項目) が表示され、ユーザは希望する要素を選択して表示を切り替えられる。

Tappy では先行研究 [42] に基づき、以下の回帰定数で

$\sigma_x^2$  と  $\sigma_y^2$  を推定する。

$$\sigma_x^2 = 0.007101W^2 + 1.412, \quad \sigma_y^2 = 0.01181H^2 + 1.365 \quad (7)$$

ここで  $W$  と  $H$  はそれぞれ  $x$  軸と  $y$  軸におけるターゲットサイズ (mm 単位) である。この先行研究では、397 人のクラウドワーカーが利き手の人差し指を使い、長方形のターゲットを合計 133,392 回タップする実験が行われた。

利用可能なオプションは以下の 3 つである。

**デバイス名:**「デバイス」のプルダウンメニューを開くと、Tappy が対応しているスマートフォンの機種の一覧が表示されるので、ユーザはその中から 1 つを選択する。Tappy は選択された機種の画面サイズ、ピクセル密度 (pixels per inch; PPI)、スケール、ユーザエージェントをシミュレートし、指定された URL にその機種でアクセスした場合のタップ可能要素を特定する。現在は iPhone のみに対応している。全ての iOS デバイスの画面サイズやピクセル密度などが既知であるため、UI 要素のサイズをピクセル単位で取得すれば mm 単位に変換可能である。

**待機時間:** ウェブページによっては、要素のレンダリングに時間がかかるため、一定時間だけ経過してから表示される要素を分析したい場合がある。このオプションで待機時間 (ミリ秒単位) を設定すると、Tappy がページの読み込みを終えてから成功率の推定を開始するまでに遅延を挟むことができる。

**JS を実行:** タップ可能要素が JavaScript によって動的に作成されるページに対応するため、このオプションはデフォルトで ON になっている。もし JavaScript が実行されるよりも前のページを分析したい場合は無効することも可能である。

#### 3.2 実装

Tappy はユーザが入力した URL と上記のオプションに基づいて、Puppeteer [47] でそのページにアクセスする。つまり Tappy は DOM を解析してページを分析するのではなく、実際にそのページをレンダリングしている。Tappy がページを読み込んで JavaScript の実行を完了した後に、全てのタップ可能な要素を取得する。タップ可能な要素には、(1) 特定の名前を持つ要素と (2) イベントリスナーを持つ要素の 2 種類がある。さらに、タップ可能な要素は (3) `<iframe>` 内にも存在する可能性がある。

(1) **特定の名前を持つ要素:** HTML においてタップ可能なオブジェクトとしてレンダリングされる要素には、`<a>`, `<button>`, `<input>`, `<select>`, `<textarea>`, `<label>` (`<input>` 用) がある。これらはタグ名を含むクエリで `querySelectorAll` [48] を使用して取得する。

(2) **イベントリスナーを持つ要素:** イベントリスナーが特定の属性 (`onclick` など) によって登録されている場合、



図 1 左：ウェブアプリ版 Tappy の画面構成。画面左側で分析対象の URL（ここでは <https://www.ipsj.or.jp/>）、対象デバイス、待機時間、JavaScript の実行の有無を設定してから「分析する」ボタンをクリックすると、画面右側に分析結果が表示される。分析画面ではタップ可能なボタンやリンクが矩形で囲まれ、タップ成功率が付記されている。矩形の色はタップ成功率（0～100%）に対応して赤から緑に変化する。中央：ある UI 要素をクリックすると、タップ成功率、大きさ（ピクセル単位、mm 単位）が表示される。画面解像度と PPI、およびスケール値によっては、要素のピクセル単位の大きさは整数にならないことがある。右：ページのトップ画像と、その画像を切り替えるためのボタン（矢印のアイコン）が重なっている。#1 をクリックするとトップ画像、#2 をクリックすると矢印アイコンの成功率と大きさが表示される。

属性名を含むクエリで要素を取得できる。ただしイベントリスナーが `addEventListener` [49] によって登録されている場合、要素が持つイベントリスナーを確認する API が存在しないため、バニラ JavaScript API では取得できない問題がある。しかし Chrome コンソール\*<sup>3</sup>では、要素が持つイベントリスナーをリストする `getEventListeners` [50] を使用でき、Puppeteer でもこのメソッドが使用可能である。これにより、Tappy は Touch Events [51]、Pointer Events [52]、および UI Events [53]（ユーザの操作に起因しない `abort`、`error`、`load`、`unload` イベントは除く）のリスナーを持つ要素を取得する。

(3) **iframe 内の要素**：広告などは HTML に `<iframe>` として埋め込まれることがある。基本的に、広告は埋め込まれたウェブページとは異なるドメインを持つため、同一オリジンポリシーにより異なるドメインのソースには JavaScript でアクセスできず、`<iframe>` 内の要素はバニラ JavaScript API では取得できない。しかし、Tappy は Chrome DevTools Protocol [54]\*<sup>3</sup>を使用することで、上記の (1) と (2) の要素が `<iframe>` 内にあっても取得可能である。

これらの手順により、Tappy は全てのタッチ可能な要素を取得する。また Tappy はスタイルプロパティ (`opacity`

や `visibility` など) を考慮するため、タップ可能な要素名だが実際には不可視でタップ不可能な場合には成功率を表示しないこととしている。こうして全てのタップ可能要素を特定したら、次に `getBoundingClientRect` [55] で要素のピクセルサイズと位置を取得し、指定された iOS デバイスにおける mm 単位のサイズと位置を計算し、式 (6) でタップ成功率を推定する。

以上の結果を、ウェブページの上端から下端までを含むスクリーンショットとともに表示する (図 1)。Tappy はページをスクロールしながらスクリーンショットを撮影していくが、スクロール中に読み込まれる処理が施されている範囲はスクリーンショットに含まれない。またスクロール後にのみ表示されるように設定された要素は含まれないが、技術的にはスクロールを一定距離ごとに繰り返し、全ての要素がレンダリングされるように一時停止しつつタップ成功率を分析するといった処理を加えることは可能である。

### 3.3 利用状況

Tappy は 2024 年 1 月 31 日にリリースされ、ここではデータを公開できる 2024 年 4 月 18 日までの利用状況について述べる。表 1 に概略を示す。ユニークな URL は 5,591 種類だが、URL の末尾にスラッシュを入れても同一サイトが表示される場合は、スラッシュの有無に関わらず 1 種類の URL に合算している。

\*<sup>3</sup> Chrome DevTools はブラウザの開発者ツール全体を指し、その一部として Chrome コンソールがある。Chrome DevTools Protocol は、Chrome ブラウザと通信するための API である。

表 1 2024年1月31日から4月18日までのウェブアプリ版 Tappy の利用状況.

URL が分析された回数	8,275
URL を分析したユニークユーザ数	3,601
分析された URL の種類	5,591

図 2 は、10 回以上分析された URL を多い順に並べたものである。この図に記載された分析回数の合計は 712 であり、一方で累計の URL 分析回数は 8,275 回 (表 1) なので、残りの 7,563 回、すなわち 90%以上は分析回数が 10 未満である。これは利用者が分析したいウェブページが多岐にわたっており、ログデータが典型的なロングテールになっていることを示している。

図 3 は、分析時に設定されたスマートフォンの機種名のうち、10 回以上だったものを多い順に列挙している。このデータの分析時に最新機種であった iPhone 15 がデフォルトで入力されていたために突出して多いが、指定可能な中で最も古い iPhone 4 など、発売が 10 年以上前の機種でのページ表示についてもまんべんなく分析されている。

X 上でも Tappy の利便性に関するポストがある [56]。また、印刷会社 [57]、デザイン会社 [58]、ウェブ制作会社 [59] など業務に Tappy を活用していることを報告しており、Tappy が幅広い業種にとって有用であることを裏付けている。

## 4. Figma プラグイン版 Tappy

Tappy プラグインを利用するには、ユーザーはプラグイン紹介ページ\*2 にアクセスして「Open in...」ボタンをクリックするか、Figma のウェブアプリまたはデスクトップ版のメニューから Actions → Plugins & widgets と進み「Tappy」と検索することでインストールできる。

プラグインを起動すると、Figma 内に小さなウィンドウが表示される (図 4)。ユーザはドロップダウンメニューから iOS デバイスを選択し、開発中のアプリやウェブページ内の UI 要素をクリックすると、その要素のサイズ (ピクセルおよび mm 単位) とタップ成功率がウィンドウ内に表示される。プラグインは JavaScript で実装されており、選択された UI 要素のサイズは、Figma が提供するプラグイン実装環境内でピクセル単位で取得される。その後の mm 単位への変換とタップ成功率の計算はウェブアプリ版と同様である。

プラグインは 2024 年 10 月 28 日時点で iPhone 15 までの iOS デバイスをサポートしている。2024 年 10 月 9 日にプラグインが公開されてから、10 月 28 日の時点で 100 名がインストールしている。

## 5. ウェブアプリ版の利用例のインタビュー

著者らの所属会社内でウェブアプリ版を公開して約 2 ヶ

月経過してから、利用歴のある社員にインタビューを実施した。なお、モデルのパラメータ値はインタビューを実施した時期からアップデートされており、現在のタップ成功率推定値と必ずしも一致しない。各ウェブサービスのスクリーンショットについても同様である。

### 5.1 利用例 1: Yahoo!トラベル

この社員は iPhone 12 mini を選択し、Yahoo!トラベル\*4 を分析した。画面上の多くの UI 要素がタップ成功率 97%以上だったのに対して、横三本線のメニューアイコンのタップ成功率が相対的に低い (58%) ことに気づいたとコメントした (図 5)。

著者らが調査したところ、メニューアイコンの部分のみがタップ可能に設定されていたが、アイコン周辺にはまだ余白があるため、そこに padding [60] を追加することで、タップ可能領域の幅と高さを 7.04 mm に拡大できる。この場合、アイコンの外観を変更せずにタップ成功率が 98% 以上になる。

### 5.2 利用例 2: Yahoo!検索

この社員は iPhone 13 を選択し、Yahoo!検索\*5 で「保険」と検索し、その検索結果画面を分析した。Yahoo!検索ではクエリに関連する広告が表示されるが、図 6 右下にある情報アイコン (i マーク) を選択すると、広告が表示される理由を説明するポップアップが出る。この社員は、情報アイコンのタップ成功率が想定より低いことを発見した (81.66%)。もしデザイナーが意図するタップ成功率があり、たとえば 95%以上にしたのであれば、情報アイコンを現状よりも 3 mm だけ拡大することで実現可能である。

### 5.3 利用例 3: Yahoo!検索

この社員は iPhone SE (第 3 世代) を選択して、Yahoo!検索で特定企業の電化製品を検索し、その検索結果画面を分析した。一般的な検索結果に加えて、Yahoo!検索ではクエリに基づいて Yahoo!ショッピングで販売中の製品を推薦する (図 7)。推薦された製品の下には、(1) この商品が「優良配送」をする店舗で販売されていること、(2) 配達日、の 2 つが記載されたボタンがあり、これらをタップすることで詳細な情報を閲覧できる。これらのボタンはこの社員が実装したが、想定よりもタップ成功率が低い (68.33%) ことを発見したとコメントした。

また、Yahoo!検索は電化製品のレビューも表示している (図 8)。同じ社員が、レビューの評価ボタンのタップ成功率が低い (74.48%) ことを発見した。

\*4 <https://travel.yahoo.co.jp>

\*5 <https://search.yahoo.co.jp>

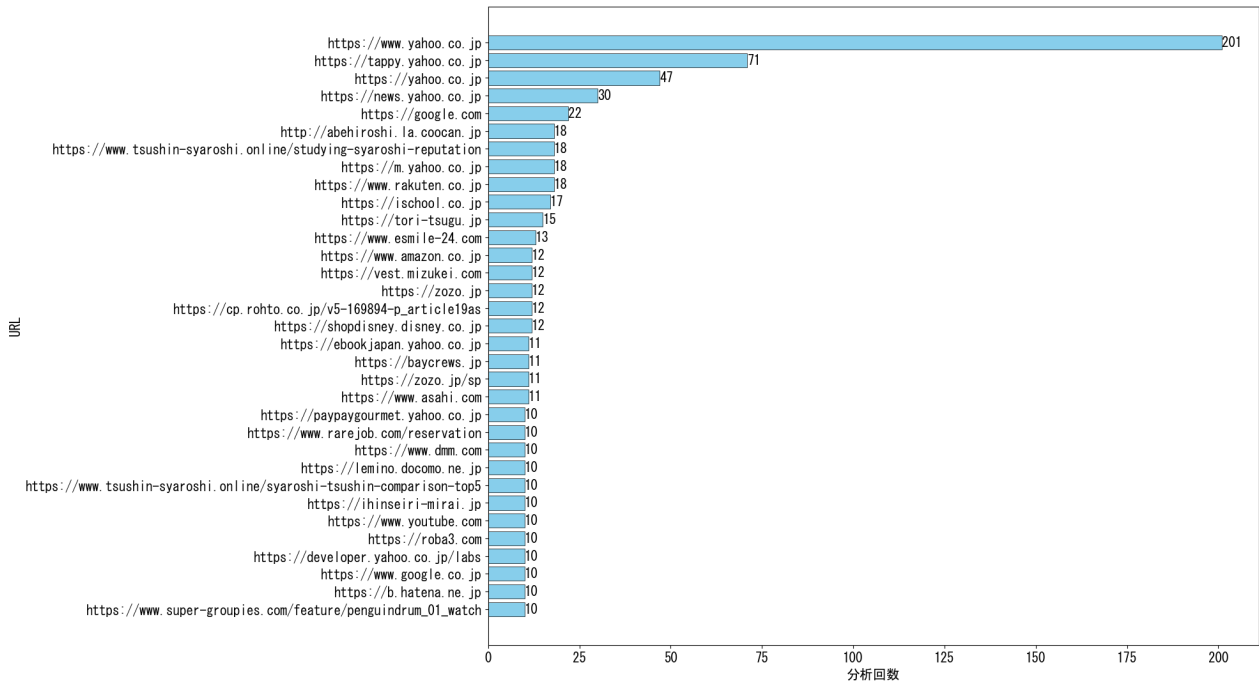


図 2 分析された URL を多い順にソートしたもの. 分析回数が 10 以上のデータに限る.

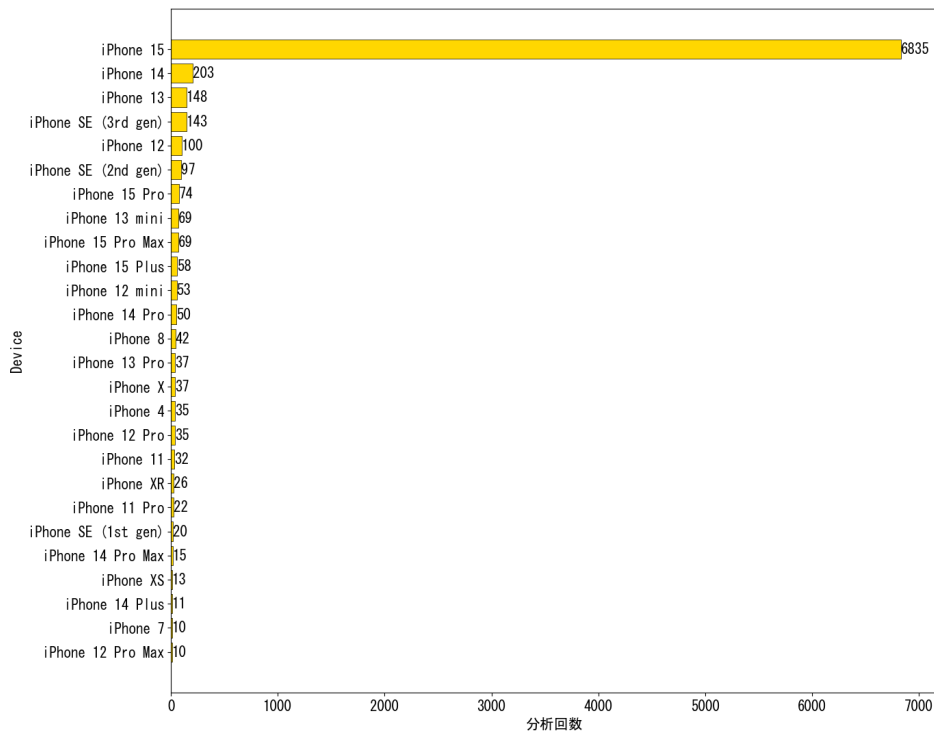


図 3 分析時に設定されたスマートフォンの回数. 分析回数が 10 以上のデータに限る.

#### 5.4 利用例 4: Yahoo! JAPAN Tech Blog

Yahoo! JAPAN Tech Blog<sup>\*6</sup>には、会社のサービス開発を支える技術やデザイン、文化について解説する記事があり、閲覧者は「学びがある」「わかりやすい」「新しい視点」の 3 つのボタンから感想を投票できる (図 9)。投票

ボタンのタップ可能領域は、タップするときだけハイライトされ、通常時は背景色と同じである。この社員は Tappy で Yahoo! JAPAN Tech Blog のページを分析したところ、「新しい視点」のボタンのタップ可能領域が他よりも縦に大きいことに気づいた。この社員は Tappy によってタップ可能領域が可視化されたことで、初めて CSS の設定ミスに気づいて実装を修正したとコメントした。これはタッ

\*6 <https://techblog.yahoo.co.jp>

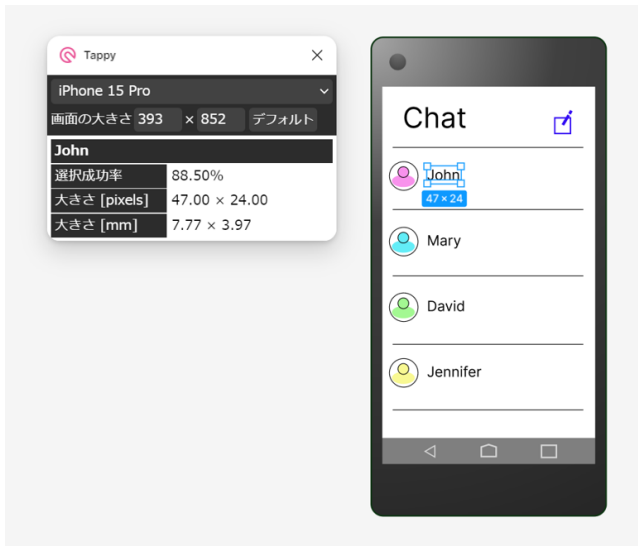


図 4 Figma プラグイン版 Tappy の画面。

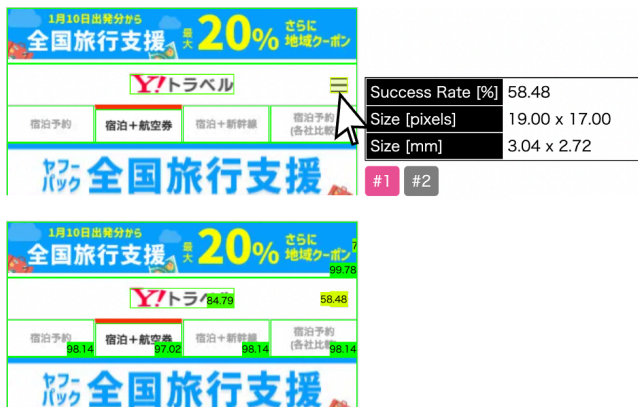


図 5 Yahoo!トラベルのトップ画面の予測結果。

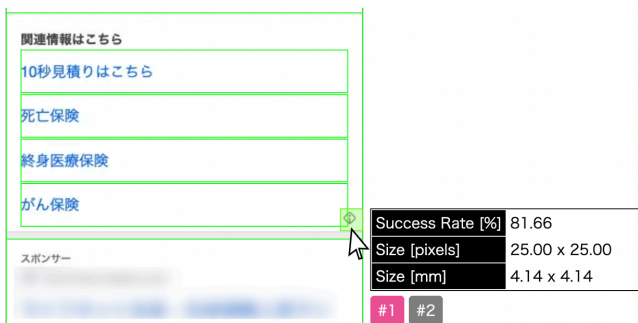


図 6 Yahoo!検索の結果に表示される情報アイコン (◇マーク)。

ブ成功率の推定モデルとは関連しないが、Tappy の可視化機能によって UI の実装修正にまで至った例である。

以上のように、Tappy で自社サービスのページを分析することで、タップ成功率が低い UI 要素が存在することに気付いた社員が複数いることが確認され、デザインを修正するきっかけを与えたことが示された。デザイナーやエンジニアが想定していたよりもタップ成功率が低くなってしまいうこともあり (利用例 2・3)、Tappy はタップ成功率の定量的な指標を提供することでデザインの再検討に貢献してい



図 7 Yahoo!検索で電化製品を検索した結果。

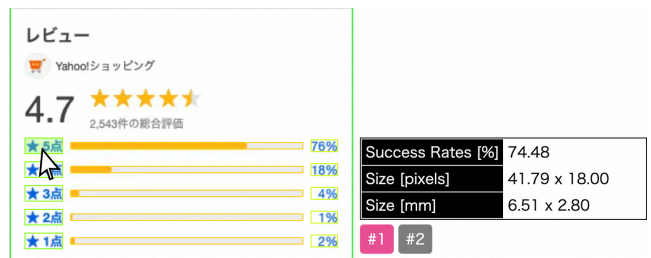


図 8 Yahoo!検索の推薦商品のレビュー画面に関する予測結果。



図 9 Yahoo! JAPAN Tech Blog の感想ボタンの変更前と後。

ると言える。また一部の社員は、Tappy が提供するタップ成功率に基づいて、開発チームのメンバーとページのデザインについて議論するのに役立ったとコメントした。よって Tappy は、定量的な議論による意思決定を促すことにも役立っており、これは従来の UI 操作モデル論文では報告されていなかった利点である。

## 6. 制約と今後の課題

Tappy のタップ成功率推定モデルには、式 (7) においてターゲットサイズ  $W$  と  $H$  に基づいてタップ座標の分散  $\sigma_x^2$  と  $\sigma_y^2$  を線形回帰で求めるプロセスが含まれる。これらの回帰定数は、クラウドワーカーが非利き手でスマートフォンを持ち、利き手の人差し指でタップするように指示したユーザ実験のデータから算出している [42]。しかしタップ座標の分散は、スマートフォンの持ち方や使用する指によって変化することが知られている [40], [61], [62]。また先行研究では、タップ成功率がユーザの年齢にも依存することが示されている [63], [64], [65]。さらに、ターゲットが画面の端にある場合 [66] や、ターゲット付近に別のボタンなどが設置されている場合 [67], [68] にもタップ成功率が変化する。現状の Tappy には、上記のような状況を考慮

してタップ成功率を補正するような機能は搭載されていない。年齢ごとの回帰分析データを適用したり、画面上の各ターゲットのレイアウトも考慮するように改良すれば、より高精度にタップ成功率を算出できるようになるだろう。

また Tappy は、実装の単純化のために UI 要素の外接矩形のタップ成功率を算出している。しかし HTML では楕円や任意形状のボタンも設置可能であり、より正確にタップ成功率を算出するのであれば CSS のプロパティや SVG の形状なども考慮する必要がある。先行研究において、円形のターゲット [6], [31] や任意形状のターゲット [69] のタップ成功率推定モデルが既に提案されているため、これらを Tappy に組み込むことでより高精度に推定できるはずである。

最後に、タップ成功率のみを考えるのであれば、可能な限りターゲットサイズを大きくするのが望ましいが、その場合は一画面内に UI 要素が入りきらずに、ウェブページが縦に長くなってしまふ。すると、ページ下部にあるリンクなどを選択したいユーザは何度もスクロールを繰り返す必要があり、その観点ではユーザビリティが低下する懸念がある [70]。現状では、ターゲットの大きさとスクロールの少なさのどちらがよりユーザビリティ向上に寄与するかは不明であり、今後の検討課題としたい。

## 7. おわりに

本稿では、スマートフォン上に表示する UI 要素の大きさに基づいてタップ成功率を推定するツール「Tappy」のウェブアプリ版と Figma プラグイン版について、理論的背景と実装、一般公開後の利用状況について述べた。UI 操作のパフォーマンスモデルに関して、従来から HCI 分野では「高精度に推定できるモデルを提供すれば、デザイナーが画面を設計するときに役立つ」と主張されることが多かった。しかし、具体的にどのようなモデルがデザイナーの業務に役立っているのかを報告した論文が見当たらず、またモデルを利用しやすいツールの形態で提供している事例が知られていなかった。我々のウェブアプリ版 Tappy は、インタビュー調査によってプロのデザイナーやエンジニアにとって有益であることを議論し、さらに一般公開から 3 ヶ月弱で 3,601 名のユニークユーザが累計 8,275 回の分析をしており、必要があることが示された。また Figma プラグイン版も、現在までに 100 名が利用している。なお、本稿で報告した利用者数や分析回数などは一般公開しているウェブアプリ版・Figma 版の Tappy であり、実際にはそれ以前から同じ機能を持つ社内向けバージョンを提供しているため、合計の利用者数・分析回数はより多くなる。今後は Figma プラグイン版についてもユーザからの定性的なフィードバックを収集し、Tappy がどのように UI デザインに役立てられているかを検証する予定である。

## 参考文献

- [1] Clark, J.: *Designing for Touch, A Book Apart*, New York, USA, 1st edition (2016).
- [2] Hooper, S. and Berkman, E.: *Designing Mobile Interfaces: Patterns for Interaction Design*, O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472, USA, 1 edition (2011).
- [3] Johnson, J.: *Designing with the Mind in Mind: Simple Guide to Understanding User Interface Design Guidelines*, Morgan Kaufmann, San Francisco, USA, 2nd edition (2014).
- [4] Neil, T.: *Mobile Design Pattern Gallery: UI Patterns for Smartphone Apps*, O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472, USA, 1 edition (2014).
- [5] Google: Touch target size (2023).
- [6] Bi, X. and Zhai, S.: Predicting Finger-Touch Accuracy Based on the Dual Gaussian Distribution Model, *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, UIST '16, New York, NY, USA, ACM, pp. 313-319 (online), DOI: 10.1145/2984511.2984546 (2016).
- [7] Yamanaka, S. and Usuba, H.: 0.2-mm-Step Verification of the Dual Gaussian Distribution Model with Large Sample Size for Predicting Tap Success Rates, *Proc. ACM Hum.-Comput. Interact.*, Vol. 8, No. ISS (online), DOI: 10.1145/3698153 (2024).
- [8] Usuba, H., Sato, J., Sasaya, N., Yamanaka, S. and Yamashita, F.: Tappy: Predicting Tap Accuracy of User-Interface Elements by Reverse-Engineering Web-page Structures (2024).
- [9] 山中祥太, 薄羽大樹, 山下郁矢, 笹谷奈翁美, 佐藤潤一: スマートフォン用ウェブページにおけるタップ成功率推定ツール Tappy の実用化, 研究報告ヒューマンコンピュータインタラクション (HCI), Vol. 2024-HCI-208, No. 25, pp. 1-7 (2024).
- [10] Figma, Inc.: Figma (2024).
- [11] Yamanaka, S., Usuba, H., Sato, J., Sasaya, N., Yamashita, F. and Yamaguchi, S.: Tappy Plugin for Figma: Predicting Tap Success Rates of User-Interface Elements under Development for Smartphones (2024).
- [12] Prazina, I., Bećirović, c., Cogo, E. and Okanović, V.: Methods for Automatic Web Page Layout Testing and Analysis: A Review, *IEEE Access*, Vol. 11, pp. 13948-13964 (online), DOI: 10.1109/ACCESS.2023.3242549 (2023).
- [13] Wu, Z., Xu, G., Lu, C., Chen, E., Zhang, Y. and Zhang, H.: Position-wise contextual advertising: Placing relevant ads at appropriate positions of a web page, *Neurocomputing*, Vol. 120, pp. 524-535 (online), DOI: 10.1016/j.neucom.2013.04.018 (2013).
- [14] Mao, J., Bian, J., Tian, W., Zhu, S., Wei, T., Li, A. and Liang, Z.: Detecting Phishing Websites via Aggregation Analysis of Page Layouts, *Procedia Computer Science*, Vol. 129, pp. 224-230 (online), DOI: 10.1016/j.procs.2018.03.053 (2018).
- [15] Grossman, T. and Balakrishnan, R.: The Bubble Cursor: Enhancing Target Acquisition by Dynamic Resizing of the Cursor's Activation Area, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '05, New York, NY, USA, ACM, pp. 281-290 (online), DOI: 10.1145/1054972.1055012 (2005).
- [16] floatas: Chrome Extension of Bubble Cursor (2018).
- [17] floatas: Implementation of the Bubble-Cursor Chrome Extension (2018).



- [18] Dixon, M. and Fogarty, J.: Prefab: Implementing Advanced Behaviors Using Pixel-Based Reverse Engineering of Interface Structure, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, New York, NY, USA, Association for Computing Machinery, p. 1525–1534 (online), DOI: 10.1145/1753326.1753554 (2010).
- [19] Dixon, M., Leventhal, D. and Fogarty, J.: Content and Hierarchy in Pixel-Based Methods for Reverse Engineering Interface Structure, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, New York, NY, USA, Association for Computing Machinery, p. 969–978 (online), DOI: 10.1145/1978942.1979086 (2011).
- [20] Dixon, M., Fogarty, J. and Wobbrock, J.: A General-Purpose Target-Aware Pointing Enhancement Using Pixel-Level Analysis of Graphical Interfaces, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, New York, NY, USA, Association for Computing Machinery, p. 3167–3176 (online), DOI: 10.1145/2207676.2208734 (2012).
- [21] Dixon, M., Nied, A. and Fogarty, J.: Prefab Layers and Prefab Annotations: Extensible Pixel-Based Interpretation of Graphical Interfaces, *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST '14, New York, NY, USA, Association for Computing Machinery, p. 221–230 (online), DOI: 10.1145/2642918.2647412 (2014).
- [22] Nguyen, T. A. and Csallner, C.: Reverse Engineering Mobile Application User Interfaces with REMAUI, *Proceedings of the 30th IEEE/ACM International Conference on Automated Software Engineering*, ASE '15, Washington, DC, USA, IEEE Press, p. 248–259 (online), DOI: 10.1109/ASE.2015.32 (2015).
- [23] Doosti, B., Dong, T., Deka, B. and Nichols, J.: A Computational Method for Evaluating UI Patterns (2018).
- [24] Schoop, E., Zhou, X., Li, G., Chen, Z., Hartmann, B. and Li, Y.: Predicting and Explaining Mobile UI Tappability with Vision Modeling and Saliency Analysis, *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, New York, NY, USA, Association for Computing Machinery, (online), DOI: 10.1145/3491102.3517497 (2022).
- [25] Liu, T. F., Craft, M., Situ, J., Yumer, E., Mech, R. and Kumar, R.: Learning Design Semantics for Mobile Apps, *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*, UIST '18, New York, NY, USA, Association for Computing Machinery, p. 569–579 (online), DOI: 10.1145/3242587.3242650 (2018).
- [26] Swearngin, A. and Li, Y.: Modeling Mobile Interface Tappability Using Crowdsourcing and Deep Learning, *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, New York, NY, USA, Association for Computing Machinery, p. 1–11 (online), DOI: 10.1145/3290605.3300305 (2019).
- [27] Li, G. and Li, Y.: Spotlight: Mobile UI Understanding using Vision-Language Models with a Focus, *The Eleventh International Conference on Learning Representations* (2023).
- [28] Wu, J., Krosnick, R., Schoop, E., Swearngin, A., Bigham, J. P. and Nichols, J.: Never-Ending Learning of User Interfaces, *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, UIST '23, New York, NY, USA, Association for Computing Machinery, (online), DOI: 10.1145/3586183.3606824 (2023).
- [29] You, K., Zhang, H., Schoop, E., Weers, F., Swearngin, A., Nichols, J., Yang, Y. and Gan, Z.: Ferret-UI: Grounded Mobile UI Understanding with Multimodal LLMs (2024).
- [30] Fitts, P. M.: The information capacity of the human motor system in controlling the amplitude of movement, *Journal of Experimental Psychology*, Vol. 47, No. 6, pp. 381–391 (online), DOI: 10.1037/h0055392 (1954).
- [31] Yamanaka, S. and Usuba, H.: Rethinking the Dual Gaussian Distribution Model for Predicting Touch Accuracy in On-Screen-Start Pointing Tasks, *Proc. ACM Hum.-Comput. Interact.*, Vol. 4, No. ISS (online), DOI: 10.1145/3427333 (2020).
- [32] Zhang, H., Huang, J., Tu, H. and Tian, F.: Shape-Adaptive Ternary-Gaussian Model: Modeling Pointing Uncertainty for Moving Targets of Arbitrary Shapes, *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, New York, NY, USA, Association for Computing Machinery, (online), DOI: 10.1145/3544548.3581217 (2023).
- [33] Huang, J., Tian, F., Li, N. and Fan, X.: Modeling the Uncertainty in 2D Moving Target Selection, *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, UIST '19, New York, NY, USA, Association for Computing Machinery, pp. 1031–1043 (online), DOI: 10.1145/3332165.3347880 (2019).
- [34] Ma, Y., Zhai, S., Ramakrishnan, I. and Bi, X.: Modeling Touch Point Distribution with Rotational Dual Gaussian Model, *The 34th Annual ACM Symposium on User Interface Software and Technology*, New York, NY, USA, Association for Computing Machinery, p. 1197–1209 (online), available from (<https://doi.org/10.1145/3472749.3474816>) (2021).
- [35] Wobbrock, J. O., Cutrell, E., Harada, S. and MacKenzie, I. S.: An Error Model for Pointing Based on Fitts' Law, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, New York, NY, USA, ACM, pp. 1613–1622 (online), DOI: 10.1145/1357054.1357306 (2008).
- [36] Bi, X., Li, Y. and Zhai, S.: FFitts Law: Modeling Finger Touch with Fitts' Law, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, New York, NY, USA, ACM, pp. 1363–1372 (online), DOI: 10.1145/2470654.2466180 (2013).
- [37] MacKenzie, I. S.: Fitts' law as a research and design tool in human-computer interaction, *Human-Computer Interaction*, Vol. 7, No. 1, pp. 91–139 (online), DOI: 10.1207/s15327051hci0701.3 (1992).
- [38] Crossman, E. R.: The measurement of perceptual load in manual operations, PhD Thesis, University of Birmingham (1956).
- [39] Welford, A. T.: *Fundamentals of skill*, Methuen Publishing, London, UK (1968).
- [40] Bi, X. and Zhai, S.: Bayesian touch: a statistical criterion of target selection with finger touch, *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST '13)*, New York, NY, USA, ACM, pp. 51–60 (online), DOI: 10.1145/2501988.2502058 (2013).
- [41] Zhu, S., Kim, Y., Zheng, J., Luo, J. Y., Qin, R., Wang, L., Fan, X., Tian, F. and Bi, X.: Using Bayes' Theorem for Command Input: Principle, Models, and Applications, *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, New York, NY, USA, Association for Computing Machinery,

- p. 1–15 (online), DOI: 10.1145/3313831.3376771 (2020).
- [42] Usuba, H., Yamanaka, S., Sato, J. and Miyashita, H.: Predicting Touch Accuracy for Rectangular Targets by Using One-Dimensional Task Results, *Proc. ACM Hum.-Comput. Interact.*, Vol. 6, No. ISS (online), DOI: 10.1145/3567732 (2022).
- [43] Yamanaka, S.: Utility of Crowdsourced User Experiments for Measuring the Central Tendency of User Performance to Evaluate Error-Rate Models on GUIs, *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, Vol. 9, No. 1, pp. 155–165 (2021).
- [44] Huang, J., Tian, F., Fan, X., Zhang, X. L. and Zhai, S.: Understanding the Uncertainty in 1D Unidirectional Moving Target Selection, *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, New York, NY, USA, Association for Computing Machinery, (online), DOI: 10.1145/3173574.3173811 (2018).
- [45] Yu, D., Liang, H.-N., Lu, X., Fan, K. and Ens, B.: Modeling Endpoint Distribution of Pointing Selection Tasks in Virtual Reality Environments, *ACM Trans. Graph.*, Vol. 38, No. 6 (online), DOI: 10.1145/3355089.3356544 (2019).
- [46] Cao, H., Lu, Y., Deng, Y., Mcfarland, D. and Bernstein, M. S.: Breaking Out of the Ivory Tower: A Large-scale Analysis of Patent Citations to HCI Research, *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, New York, NY, USA, Association for Computing Machinery, (online), DOI: 10.1145/3544548.3581108 (2023).
- [47] Puppeteer: puppeteer/puppeteer: Node.js API for Chrome (2023).
- [48] MDN contributors: Document: querySelectorAll() method - Web APIs — MDN (2023).
- [49] MDN contributors: EventTarget: addEventListener() method - Web APIs — MDN (2023).
- [50] Basques, K. and Emelianova, S.: Console Utilities API reference (2015).
- [51] MDN contributors: Touch events - Web APIs — MDN (2023).
- [52] MDN contributors: Pointer events - Web APIs — MDN (2023).
- [53] MDN contributors: UI Events - Web APIs — MDN (2023).
- [54] ChromeDevTools: Chrome DevTools Protocol (2023).
- [55] MDN contributors: Element: getBoundingClientRect() method - Web APIs — MDN (2023).
- [56] X: ウェブアプリ版 Tappy に関するポスト (2024). [https://x.com/search?q=https%3A%2F%2Ftappy.yahoo.co.jp%2F&src=typed\\_query&f=live](https://x.com/search?q=https%3A%2F%2Ftappy.yahoo.co.jp%2F&src=typed_query&f=live).
- [57] ラクスル株式会社: X におけるポスト (2024). <https://x.com/satomium1/status/1765116250789982678>.
- [58] 株式会社グッドパッチ: 新しいものが大好きな Goodpatch で 2 月話題になったアプリ、サービス、デザインまとめ (2024) (2024). <https://goodpatch.com/blog/2024-02-monthly-topic>.
- [59] 株式会社 ネットインパクト: タップ操作の成功率を算出してくれる「Tappy」でアクセシビリティ検証が捗るかも (2024). <https://www.netimpact.co.jp/diary/25307/>.
- [60] MDN contributors: padding - CSS: Cascading Style Sheets — MDN (2023).
- [61] Azenkot, S. and Zhai, S.: Touch Behavior with Different Postures on Soft Smartphone Keyboards, *Proceedings of the 14th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '12, New York, NY, USA, Association for Computing Machinery, p. 251–260 (online), DOI: 10.1145/2371574.2371612 (2012).
- [62] Wang, F. and Ren, X.: Empirical evaluation for finger input properties in multi-touch interaction, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, New York, NY, USA, Association for Computing Machinery, p. 1063–1072 (online), DOI: 10.1145/1518701.1518864 (2009).
- [63] Findlater, L., Froehlich, J. E., Fattal, K., Wobbrock, J. O. and Dastyar, T.: Age-Related Differences in Performance with Touchscreens Compared to Traditional Mouse Input, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, New York, NY, USA, Association for Computing Machinery, p. 343–346 (online), DOI: 10.1145/2470654.2470703 (2013).
- [64] Findlater, L. and Zhang, L.: Input Accessibility: A Large Dataset and Summary Analysis of Age, Motor Ability and Input Performance, *Proceedings of the 22nd International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '20, New York, NY, USA, Association for Computing Machinery, (online), DOI: 10.1145/3373625.3417031 (2020).
- [65] Hourcade, J. P. and Berkel, T. R.: Simple pen interaction performance of young and older adults using handheld computers, *Interacting with Computers*, Vol. 20, No. 1, pp. 166–183 (online), DOI: 10.1016/j.intcom.2007.10.002 (2008).
- [66] Usuba, H., Yamanaka, S. and Sato, J.: Clarifying the Effect of Edge Targets in Touch Pointing through Crowdsourced Experiments, *Proc. ACM Hum.-Comput. Interact.*, Vol. 7, No. ISS (online), DOI: 10.1145/3626469 (2023).
- [67] Yamanaka, S.: Effect of Gaps with Penal Distractors Imposing Time Penalty in Touch-pointing Tasks, *Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '18, New York, NY, USA, ACM, (online), DOI: 10.1145/3229434.3229435 (2018).
- [68] Yamanaka, S.: Risk Effects of Surrounding Distractors Imposing Time Penalty in Touch-Pointing Tasks, *Proceedings of the 2018 ACM International Conference on Interactive Surfaces and Spaces*, ISS '18, New York, NY, USA, ACM, pp. 129–135 (online), DOI: 10.1145/3279778.3279781 (2018).
- [69] Zhang, Z., Huang, J. and Tian, F.: Modeling the Uncertainty in Pointing of Moving Targets with Arbitrary Shapes, *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI EA '20, New York, NY, USA, Association for Computing Machinery, p. 1–7 (online), DOI: 10.1145/3334480.3382875 (2020).
- [70] Schwab, M., Hao, S., Vitek, O., Tompkin, J., Huang, J. and Borkin, M. A.: Evaluating Pan and Zoom Timelines and Sliders, *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, New York, NY, USA, Association for Computing Machinery, pp. 1–12 (online), DOI: 10.1145/3290605.3300786 (2019).