

# 人間の記憶容量と認知モードに対応したペン型デバイスの 拡張による GUI 非依存の描画作業支援システム

羽木 貴昭<sup>1</sup> 田野 俊一<sup>1</sup> 橋山 智訓<sup>1</sup> 市野 順子<sup>1</sup> 岩田 満<sup>2</sup> 高野 健太郎<sup>1</sup>

**概要:** 一般的に広く使用されている GUI をベースとした描画作業支援システムでは, GUI メニューの操作によって機能の切り替えやパラメータの調整を行う. この操作は, 描画作業を一時的に中断しなければならず, 認知的負荷の増加が懸念される. また, ソフトウェアの高機能化によって GUI メニューの構造はより複雑になり, 認知的負荷の増加はより深刻になることが予想される. この問題を解決するために, 2種類の実験から得た記憶容量と認知モードという2つの設計指針をもとにして, ペン型デバイスの拡張によって実現される GUI 非依存の描画作業支援システムの設計と実装を行った.

## A GUI Independent Drawing Support System using an Augmented Pen Device that Corresponds to the Human Memory Capacity and Cognitive Mode

HAGI TAKAAKI<sup>1</sup> TANO SHUNICHI<sup>1</sup> HASHIYAMA TOMONORI<sup>1</sup> ICHINO JUNKO<sup>1</sup> IWATA MITSURU<sup>2</sup>  
TAKANO KENTAROU<sup>1</sup>

**Abstract:** Users have to operate on the GUI menu to change functions or adjust parameters when using the GUI based drawing support systems which are widely used nowadays. It is concerned that these menu operations will interrupt the process of drawing and increase the cognitive load. Furthermore, GUI menus are becoming more and more complicated with the development of high performance paint software, which is concerned to cause the problem of cognitive load to get worse. To resolve this problem, on the basis of 2 design guides, memory capacity and cognitive mode, which are resulted from our experiments, we designed and implemented a GUI independent drawing support system using an augmented pen device.

### 1. はじめに

広く一般的に使用されている描画作業用のソフトウェアでは, 図 1 のように GUI メニューが備わっている. GUI メニューは, ブラシツールや消しゴムツールなどの機能を切り替えることや, ブラシの色や太さのような連続的なパラメータの調整を行うことが主な役割である. 利用者は, 描画作業に伴うこれらの操作を, GUI メニューを介して行う.

しかし, GUI メニューはポインティングデバイスによ

るクリックやドラッグなどのポインティング操作によって使用されるため, 同じくポインティング操作によって行わ

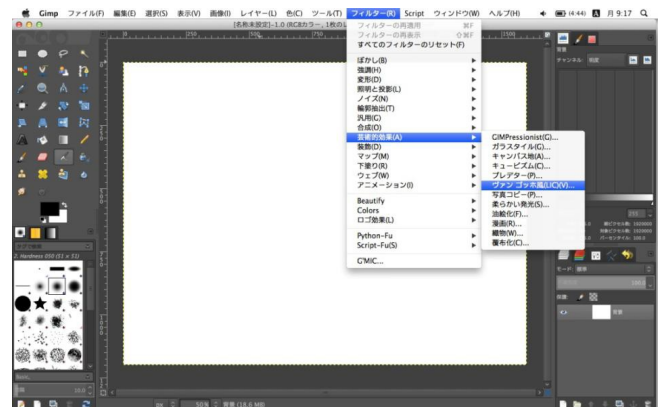


図 1 GIMP(GNU Image Manipulation Program)

<sup>1</sup> 電気通信大学 情報システム学研究科  
University of Electro-Communications, The Graduate School of Information Systems

<sup>2</sup> 東京都立産業技術高等専門学校  
Tokyo Metropolitan College of Industrial Technology

れる描画作業をその都度中断しなければならない。また、GUIメニューは視覚中心に操作されるユーザインタフェースであるため、描画対象から視線を外しメニューに注視することを強いられるため、作業パフォーマンスの低下や認知負荷の増加が懸念される [1]。

## 2. 研究の背景と問題意識

描画作業を支援するソフトウェアでは、GUIメニューを用いて以下の2つを行っている。

- (1) 機能の選択・切り替え
- (2) パラメータの調整

これらを行うためのGUIメニュー操作は、非常に煩雑であり作業パフォーマンスの低下を招くと関連研究で示唆されている。また、これらの操作をGUIメニュー操作ではなく、別の手法で実現して利便性の向上を図っている研究がいくつか行われている。特に、素早く精緻な描画が可能であることから描画作業と相性が良いとされるポインティングデバイスであるペン型デバイスに、副次的なチャンネルを加えることで拡張している研究が多い。上記の2つの操作を行うには、それぞれ、

- (1) 離散的な副次的チャンネル
- (2) 連続的な副次的チャンネル

が必要となる。

## 3. 関連研究とその問題点

Vignette[2] は、スケッチ作業に必要なパターンテクスチャを自動生成するもので、利用者の手書き操作によってパターンが生成されるため、機械的なテクスチャではなく利用者が手で描いたようになることが特徴である。ただし、あくまでテクスチャに限った機能であり、自動生成であるため詳細なパラメータの調整は行えないという問題点が見られる。

また、GUIメニュー構造に関する研究に CommandMap[3]があり、GUIの操作効率が問題視されている。この研究では階層的なGUIメニューよりも画面全体に配置された階層の無いGUIメニューのほうが作業パフォーマンスが高いことを示している。しかし、GUIメニューを使う時点でマウスカーソルの移動に依存するので、ポイン

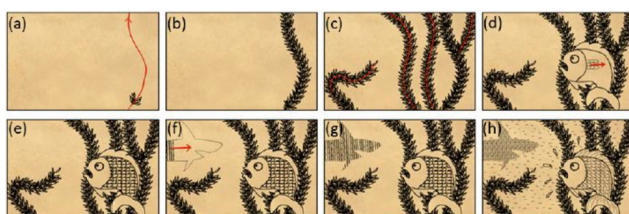


図 2 Vignette

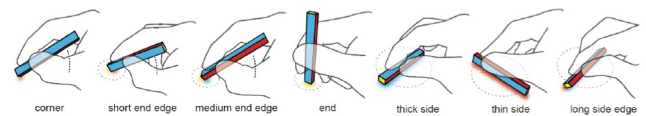


図 3 Conté

ティングによるコマンド指定には限界がある。

本研究では、GUIメニュー操作に伴う認知的な副作用を排除し、機能性を保つためのアイデアとしてペン型デバイスの拡張に注目している。

ペン型デバイス拡張によるモード切替手法の研究の一つに、Conté[4]がある。これは直方体のデバイスで、ペン型デバイスの亜種とみることができる。直方体という形状であるため、8頂点、12辺、6面を持つ。タッチディスプレイに接触させる箇所に応じたモード(機能)に変更させることで、自然で素早いモード切替の実現を目指したものである。

A-Coord Input[5]は、ペン軸中心の角度、ペン軸とタッチディスプレイがなす角度の2つの角度パラメータと筆圧を、描画やコマンド選択のパラメータに対応させ、描画とパラメータ調整を同時に行う手法を提案した研究である。

Grips and Gestures on a Multitouch-Pen[6]では、ペン型デバイスの筒部分(手で握る部分)にタッチセンサーを設置し、持ち方を識別できるようにしている。この研究では、描画に用いられる道具は様々な持ち方があることに注目しており、握り方に応じてモードを自動変更し、自然なモード切替を行うことを提案している。

HoverWidget[7]はペン型デバイスでの新たなジェスチャー入力手法を提案している。ペン型デバイスは軽

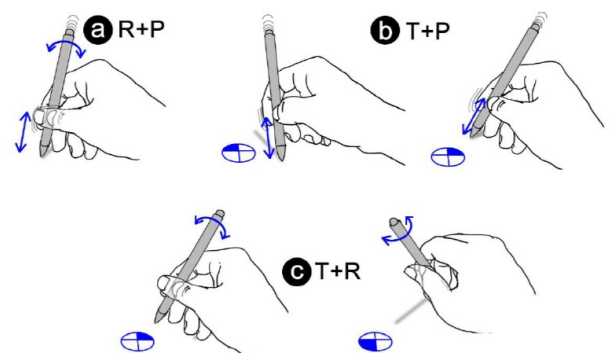


図 4 A-Coord Input

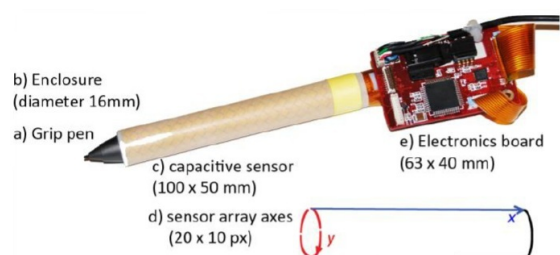


図 5 Grips and Gestures on Multitouch Pen

量で手に持って操作するため、マウスのように机上に常に設置するものではなく、タッチディスプレイから上方へ浮かせた状態 (Hover) が存在する。その状態でペンの動きをコマンド入力に対応付けることで新たなジェスチャー入力手法を実現している。

VoicePen[8] では、利用者の声を描画のパラメータ調整に利用する手法を提示している。音声で線の太さや色などを変化させることができるため、描画と同時並行のパラメータ調整が可能となる。

また従来のペン型デバイスによく見られるコマンド入力などの操作手法に、筒部分に設置されたボタン (Barrel Button), 振り動作による操作 (Jerking Movement), 長押しによる選択 (Press and Hold), ペン先の圧力によるパラメータ調整 (Pressure Based), ペン先と逆の端の利用 (Eraser End of a Pen) などがある。

しかし、関連研究において、離散的な副次的チャンネルと連続的な副次的チャンネルの両方を備えているものは無い。連続的なチャンネルに複数の閾値を設定して離散的なチャンネルとする手法もあるが、視覚的な補助が必要であったり、連続的なチャンネルが使用できなくなるなどの問題点がある。

本研究では、離散的な副次的チャンネルと連続的な副次的チャンネルの両方を備えたペン型デバイスを製作し、それを中心として GUI メニューへの依存度を軽減する描画作業支援のシステムの設計と実装を行った。また、システムの設計と実装に伴って、GUI メニュー操作による悪影響の調査と描画作業における機能の選択・切り替えの傾向の調査を目的とした2つの実験を行った。

#### 4. 本研究のアプローチ

本研究では、まず GUI メニューの及ぼす悪影響に関して調査する実験を行った。関連研究では、GUI メニュー操作が煩雑であることや作業パフォーマンスの低下を指摘することに留まっており、認知的な側面に関する調査は行われていない。本研究では特に、記憶容量に注目した。描画作業とは、頭の中のアイデアを具体化させる作業であるため、短期記憶への悪影響は致命的であり、GUI メニューが記憶容量に悪影響を及ぼしている場合、GUI メニューへの依存度を軽減することの必要性が明確になると考えた。

次に、GUI メニューではない別の手法での機能の選択・切り替えやパラメータの調整を行う場合、問題となるのは切り替え可能な機能の数である。GUI メニューは構造が非常に柔軟で膨大な機能を扱えるため、それと比較した場合、ペン型デバイスに副次的チャンネルを加えることで可能となる切り替えの数は非常に少ない。数少ない切り替えの数で、実際の描画作業での利用に有効かどうかを調査するため、機能の切り替え頻度やその分布を調査する実験を行った。

また、これらの実験をふまえて、柔軟物体で構成されたペン型デバイスの開発とそれを中心とした描画作業支援シ

ステムの設計と実装を行った。

### 5. 実験1: GUI メニュー構造が記憶容量に及ぼす影響の調査

#### 5.1 実験の目的

描画作業の、特に試行錯誤を伴う初期段階では、利用者は様々なアイデアを頭の中に浮かべながら描画を行う。その際、煩雑な GUI メニュー操作によって機能の切り替えを行うことで認知負荷が生じ、結果として重要なアイデアが失われているのではないかと考え、定量的に調査した。

#### 5.2 実験手順

本実験の被験者は22~24歳の男性10名である。被験者には、まず課題画像を覚えてもらい、その直後に GUI メニューを備えたドローイングソフトで課題画像を見ずに再現するタスクを課した。ここでは課題画像を実際の描画作業における頭の中のアイデアの代わりとしている。

課題画像には図7に示すような簡易的な地図を使用し、図6の液晶タブレットに30秒間提示した。課題画像は道を表す線と地図記号で構成されている。道には太さ、色、位置、地図記号には種類、位置の要素があり、課題画像はこれらの要素の数によって難易度が設定されている。要素が多い課題画像ほど難易度が高いとした。

ドローイングソフトには、図8に示すように、階層的なリスト構造の GUI メニューを備えているものを使用した。この GUI メニューは、階層の数が違うものを何種類か用意した。なお、ドローイングソフトでは使用できる機能は全て同じである。

本実験では、1階層、3階層、5階層の3種類の異なる階層数の GUI メニューを備えたドローイングソフトと、3種類の異なる難易度の課題画像を使用した。

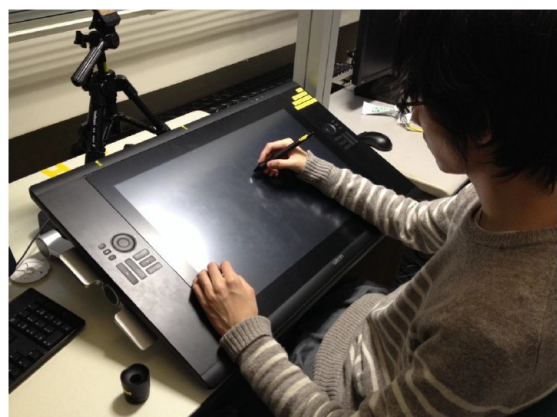


図6 Wacom Cintiq 24HD

#### 5.3 実験結果

実験結果を図9に示す。この結果は課題画像の難易度とドローイングソフトの GUI メニューの階層数の組み合わせ



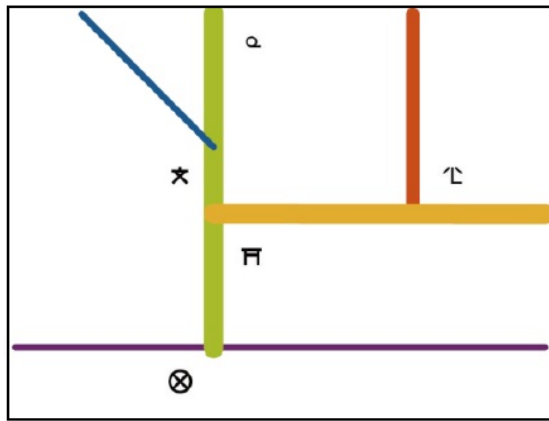


図 7 課題画像の例 (難易度 2)



図 8 ドローイングソフトの GUI メニュー

せごとの、再現率の平均値を表している

この結果から、GUI メニューの階層数が増加するほど、再現率が低下している。また、各難易度における階層数 1 と階層数 5 の 3 組に対して t 検定を行ったところ、いずれにおいても有意差があることが分かった ( $p < 0.05$ )。

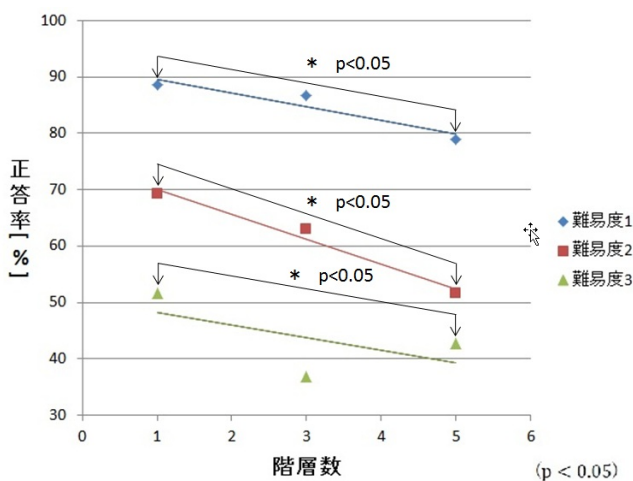


図 9 実験結果

#### 5.4 考察とまとめ

結果から、GUI メニューの階層が増えるほど再現率が低下しているため、GUI メニュー操作によって短期記憶が失われることが判明した。また、階層の数が増加するにした

がってこの悪影響は増大しているため、多くの機能を備えたソフトウェアではより顕著であると言える。しかしながら、GUI メニューの階層数を減らすと、メニュー領域が画面を覆い尽くしてしまい、狭い描画領域で作業をしなければならない。

以上のことから、GUI メニューだけに頼った機能の選択・切り替えでは、作業パフォーマンスだけでなく、記憶容量においても悪影響を及ぼし、描画作業を通じて出来る上がるデザインなどの品質を大きく損なうと考えられ、別の機能選択・切り替え手法の必要性が明確になった。

## 6. 実験 2：描画作業中の機能切り替えの分布の調査

### 6.1 実験の目的

描画作業で使用される機能群は膨大であるのに対して、ペン型デバイスの拡張によって実現される切り替え手法では、扱える機能の数は非常に少ない。この点から、GUI メニューの代替手法として有効であるかどうか疑問が残る。

そこで本実験では、描画作業における機能の選択・切り替えの分布を調査し、機能の使用頻度や切り替えが生じるタイミングの傾向などからペン型デバイスの拡張によって実現される切り替え手法が有効かどうかを検討した。また、この実験から得られた結果を描画作業支援システムの設計にも生かすことも目的とした。

### 6.2 実験手順

本実験の被験者は 23 歳の男性 1 名、女性 1 名の合計 2 名である。被験者には約 90 分間、提示した課題に沿って描画作業を行ってもらい、その際に生じた機能の選択・切り替えのタイミングと選択した機能を計測した。

提示した課題は以下の 2 つである。

- (1) 『女子小・中学生向けのニューイヤーカードのデザイン』
- (2) 『少年漫画雑誌の表紙のデザイン』

### 6.3 実験結果

実験結果を図 10、図 11 に示す。また、実験結果から、各機能の使用回数を使用頻度の高いものから昇順に並び替えたものを図 12 に示す。

図 12 から、使用頻度が極端に高い機能が見られる。これらは 10 個程度と非常に少数であり、機能切り替え回数における使用頻度の上位 10 個が占める割合は、平均で 83.1%であった。

次に、1 分間当たりの各機能の使用回数を図 13 に示す。このグラフからは、常に高頻度で使用されている機能の他に、一時的に高頻度になるバースト性を持った機能が見られた。また、これらの機能の中には、複数の機能が同時に高頻度になるところがあった。

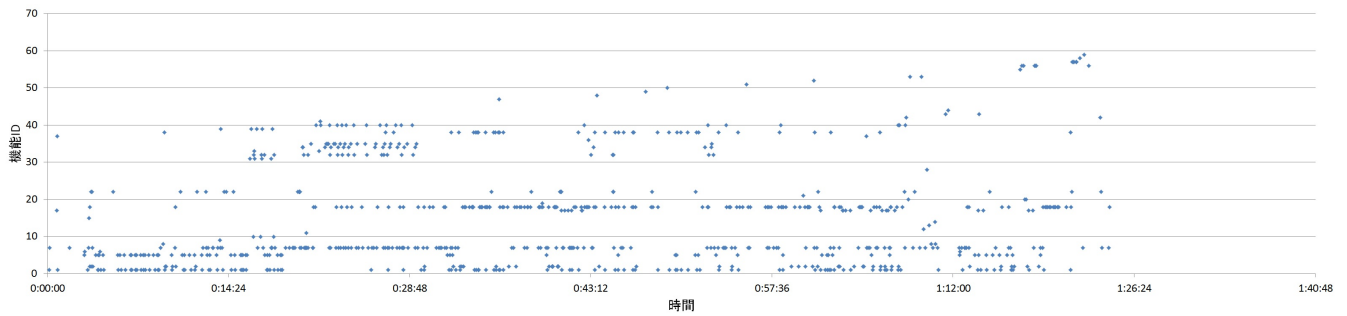


図 10 実験結果 (被験者 1)

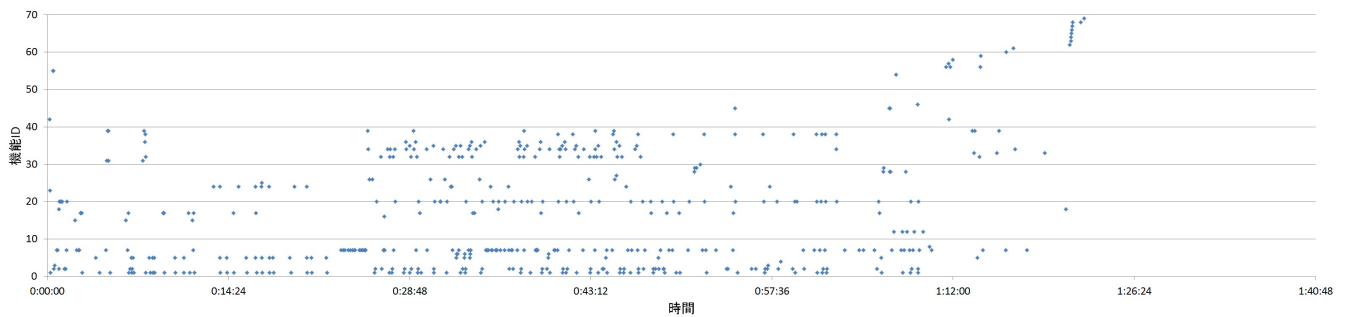


図 11 実験結果 (被験者 2)

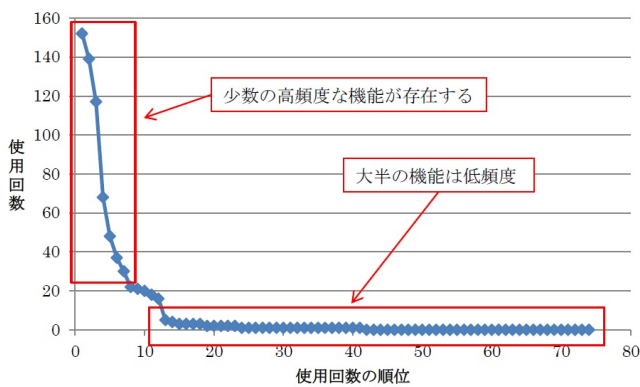


図 12 各機能の使用回数

次に、常時高い頻度で使用されてはいなくても、一時的に使用頻度が上昇するバースト的な機能があることがわかった。これらの機能群は、一時的に描画作業の中心となるので、これらの機能の使用を支援する仕組みが必要であると考えられる。また、バースト的な機能群には、複数の機能が同時に高頻度で使われる傾向が見られた。これは、複数の機能の間で頻繁に切り替え、機能を組み合わせることで作業を行っていたためだと考えられる。そのため、これらの機能を素早く切り替えられる手法は作業パフォーマンスの向上に有効である。

また、実験を通して、描画ソフトの機能群を以下の3つに区分した。システムには、それぞれに適したユーザインタフェースが必要であると考えられる。

- (1) 高頻度な機能
- (2) バースト的な機能
- (3) 低頻度な機能

## 7. システムの設計

本システムの全体構成を、図 14 に示す。実験 1 から、GUI メニュー操作による機能の選択・切り替えでは認知的にも悪影響が生じてしまうことが明確になった。そのため、実験 2 の結果より区分した高頻度な機能とバースト的な機能では、GUI メニューではなく、ペン型デバイスの拡張による手指操作での素早く視覚に依存しない切り替え手法を用いるように設計した。また、低頻度な機能については、使用回数が低いため悪影響は大きくないと考え、一時的に GUI メニューを表示するようにした。

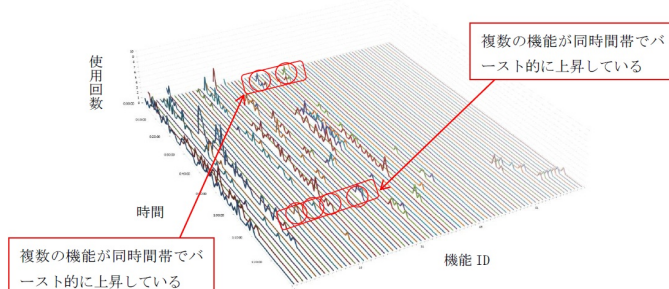


図 13 1 分間当たりの使用回数

### 6.4 考察とまとめ

実験結果から、全描画作業を通して使用頻度の高い機能は 10 個程度であることがわかった。また、それらは機能切り替えの約 8 割を占めており、これら高頻度な機能の選択・切り替えを行う手法でも十分に支援には有効であると考えられる。

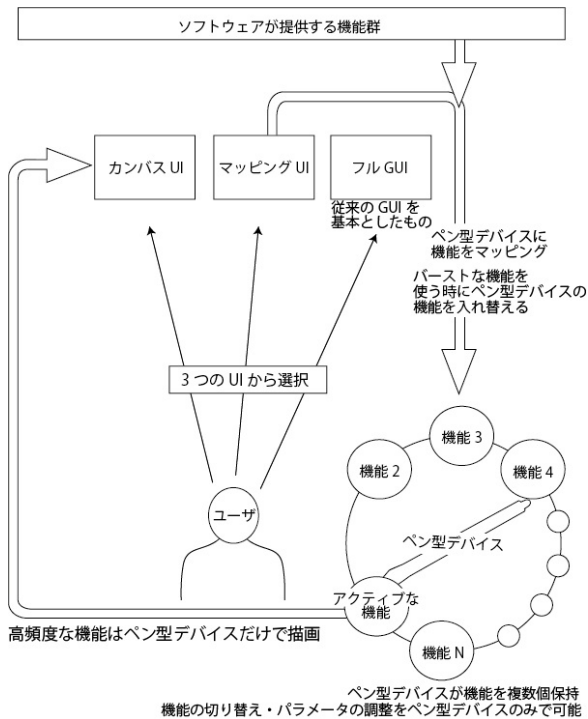


図 14 システムの全体構成

次に、ペン型デバイスの拡張によって実現される切り替え手法は、ペン型デバイスにいくつかの機能をマッピングして切り替えを行うが、その数は少数であるため、通常は高頻度な機能のみをマッピングしている。パースト的な機能を効率的に使用するために、一時的にペン型デバイスにマッピングされている機能を入れ替えることが可能な可変的マッピングが必要であると考えた。本システムでは、ペン型デバイスに任意の機能をマッピングするためのユーザーインターフェースを用意した。

## 8. ペン型デバイスの拡張方法

ペン型デバイスに副次的チャンネルを付与して拡張することで、利便性の向上を図る研究はいくつか行われているが、GUIメニューが担っている以下の2つの役割を直接行える拡張方法を提案している研究は無い。

- (1) 機能の選択・切り替え (離散的な副次的チャンネル)
- (2) パラメータの調整 (連続的な副次的チャンネル)

そこで、本システムでは、柔軟物を握る動作に着目し、握り方と握り強さによる操作手法を提案する。関連研究 [9] においても、握る動作による操作の有効性が示されているが、握るという動作は、指や手のひらの姿勢・位置によって決まる「握り方」と、各指にどれだけの力を加えるかの「握り強さ」の要素から成り立っており、それぞれが離散的な副次的チャンネルと連続的な副次的チャンネルとして利用できる。

本システムのペン型デバイスの拡張方法では、ペン型デ

バイスを柔軟な材料で構成し、その材料の形状をセンシングすることで、握り方と握り強さを測定し、副次的チャンネルとして利用する。

### 8.1 材料形状のセンシング

ペン型デバイスを構成する柔軟材料として、ポリウレタンの合成スポンジを使用した。スポンジをペン型デバイスのグリップ部分に取り付けた。また、このスポンジの形状をセンシングするために、図 15 のようにペン型デバイスの軸中心部分に反射型光センサであるフォトリフレクタをアレイ状に設置した。関連研究 [10] においても、フォトリフレクタによる柔軟物の形状のセンシングが有効であると述べられている。今回試作したペン型デバイスでは、フォトリフレクタを軸中心に4つ並べたものを軸方向に4つ設

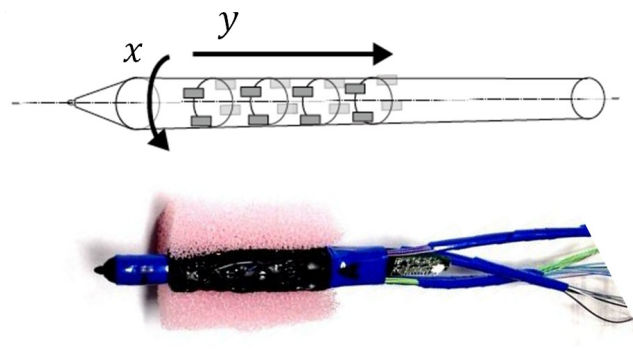


図 15 ペン型デバイスの概要図

### 8.2 握り方の識別

握り方の識別は、ペン型デバイスの軸部分に取り付けられているフォトリフレクタアレイの信号パターンを、あらかじめ登録されている複数の握り方の信号パターンと比較して、相関度を計算することで行った。相関度の計算は以下の式を使用した。

$$h_{(m_i)} = \frac{1}{\sum_{i=1}^{16} (m_i - n_i)^2} \quad (1)$$

各握り方との相関度を計算し、最も相関度が高い握り方を現在の握り方だと識別する。なお、 $h_{(m_i)}$  は相関度、 $m_i$  は各センサの信号値、 $n_i$  はあらかじめ登録されている各センサの信号値、 $i$  はフォトリフレクタの番号である。図 16 に、フォトリフレクタアレイの信号パターンを可視化したものを示す。

また、16個のフォトリフレクタの信号値のうち、値が大きいもの3つを連続的な副次的チャンネルとして使用した。これによって、握り方に応じた握り強さの測定が行える。

## 9. デモシステム

### 9.1 システムの実装

本システムは、ペン型デバイスによる GUI メニューに依



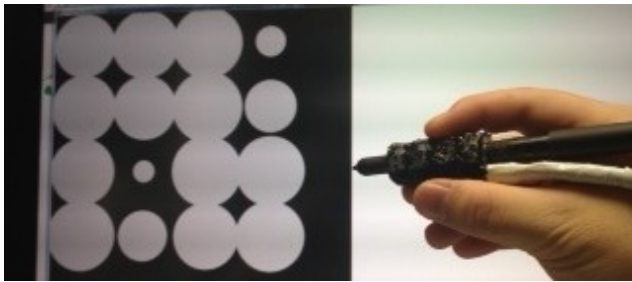


図 16 フォトリフレクタアレイの信号パターンの可視化

存しない機能の選択・切り替えやパラメータ調整を行え、高頻度な機能、バースト的な機能、低頻度な機能に合わせた3つのユーザインタフェースを備えた描画作業支援システムである。本研究では、このシステムを実装した。

実装には、2Dグラフィックスライブラリ Cairo を使用した。また、ペン型デバイスとの通信に Arduino と Firmata ライブラリを使用した。さらに、低頻度な機能のために一時的に使用する GUI メニューの作成には、Gtk ライブラリを使用した。

## 9.2 ユーザインタフェース

図 17, 図 18, 図 19 にデモシステムの3つのユーザインタフェースを示す。図 17 がキャンバス UI, 図 18 がマッピング UI, 図 19 がフル GUI である。

キャンバス UI は、画面全体が描画領域になっており、ブラシツールや消しゴムツールなど、高頻度な機能をペン型デバイスによって操作する。また、左上にマッピング UI とフル GUI を起動するボタンのみが設置されており、バースト的な機能や低頻度な機能を使用する時にだけこのボタンから別のユーザインタフェースを起動する。

バースト的な機能を使用する時には、キャンバス UI の左上にあるマッピング UI 起動ボタンを押して、マッピング UI に切り替える。マッピング UI は図 18 のように機能が一覧表示されている。ペン型デバイスには、握り方ごとに機能がマッピングされているが、ある握り方でペン型デバイスを持ち、マッピング UI 上の機能を選択すると、その握り方によって起動する機能が選択したものに置き換わり、使用できるようになる。

最後に、キャンバス UI にあるフル GUI のボタンを押すと、標準的な GUI メニューを備えたユーザインタフェースに切り替わり、使用できるようになっている。

## 10. まとめと今後の課題

本研究では、一般的に使用されている描画作業用のソフトウェアに用いられている GUI メニューの操作の煩雑さに注目し、それが描画作業中の利用者に与える悪影響を低減する描画作業支援システムの構築を目指した。

システムの設計に向けて、描画作業支援を目的とした関連研究の調査と、2つの実験を行った。この実験から、GUI

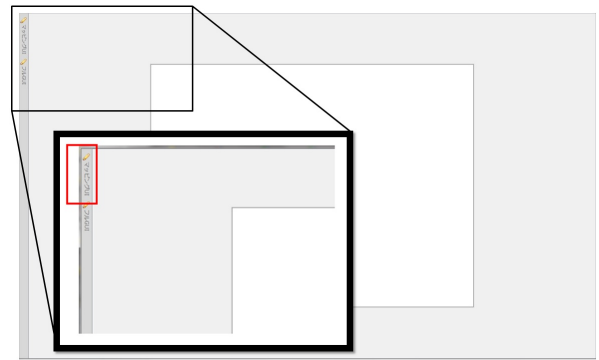


図 17 キャンバス UI

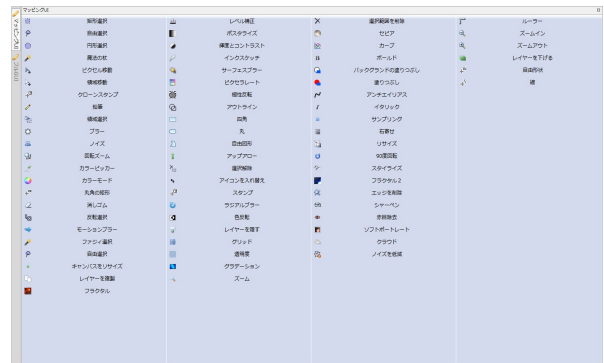


図 18 マッピング UI

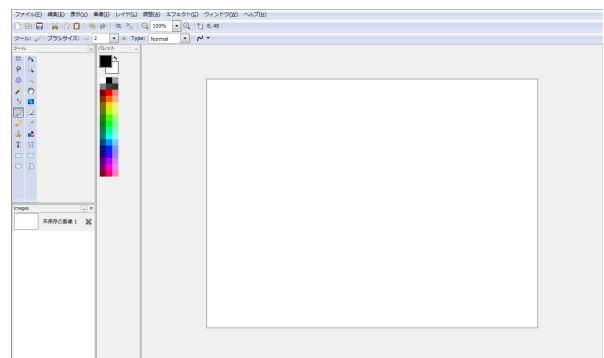


図 19 フル GUI

メニューの階層構造が深くなるほど覚えていられる記憶の量が減少することが確かめられた。また、使用頻度が高い機能はごく少数であることがわかった。さらに、作業全体では高頻度ではなくても、一時的に高頻度になるバースト的な機能があり、これらの機能は複数を組み合わせて使用されることから、同時に複数の機能が高頻度になる様子が見られた。

関連研究から、以下が要件が必要であると考えた。

- (1) 描画と並行したパラメータ調整
- (2) 離散的な副次的チャンネル (機能の選択・切り替え)
- (3) 連続的な副次的チャンネル (パラメータの調整)
- (4) 機能の切り替えとパラメータの調整を同時に操作可能

本研究では、握り動作に注目し、光センサアレイによる握り方の識別手法を用いて、ペン型デバイスの柔軟材料に

よる拡張を行った。柔軟材料の形状の変化を光センサアレイによってセンシングすることで、握り方の識別と握り強さの測定を行い、機能の切り替えとパラメータの調整を行えるペン型デバイスの開発を行った。

以上をふまえて、ペン型デバイスによる機能の選択・切り替えとパラメータの調整が可能な、キャンバス UI・マッピング UI・フル GUI という 3 つの UI を備えた非 GUI を中心とする描画作業支援システムの設計と実装を行った。

今後の課題として、以下の 2 点が考えられる。

- (1) 様々な GUI コンポーネントを用いた記憶容量に関する実験の実施
- (2) ペン型デバイスの握り方識別の精度向上

## 参考文献

- [1] 羽木貴昭, 田野俊一, 橋山智訓, 市野順子, 岩田満, 高野健太郎: ペン型デバイス拡張による GUI 非依存の描画支援システムの提案, ヒューマンインタフェースシンポジウム 2013, pp.883-886, (2013).
- [2] Kazi, R. H., Igarashi, T., Zhao, S. and Davis, R.: Vignette: Interactive Texture Design and Manipulation with Freeform Gestures for Pen-and-ink Illustration, Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp.1727-1736, (2012).
- [3] Scarr, J., Cockburn, A., Gutwin, C. and Bunt A.: Improving Command Selection with CommandMaps, Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp.257-266, (2012).
- [4] Vogel, D. and Casiez, G.: Conté: Multimodal Input Inspired by an Artist's Crayon, Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology, pp.357-366, (2011).
- [5] Hasan, K., Yang, X., Bunt, A. and Irani, P.: A-coord Input: Coordinating Auxiliary Input Streams for Augmenting Contextual Pen-based Interactions, Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp.805-814, (2012).
- [6] Song, H., Benko, H., Guimbretiere, F., Izadi, S., Cao, X. and Hinckley, K.: Grips and Gestures on a Multi-touch Pen, Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp.1323-1332, (2011).
- [7] Grossman, T., Hinckley, K., Baudisch, P., Agrawala, M. and Balakrishnan R.: Hover widgets: using the tracking state to extend the capabilities of pen-operated devices, Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp.861-870, (2006).
- [8] Harada S., Saponas, T. S., and James, A. L.: VoicePen: Augmenting Pen Input with Simultaneous Non-linguistic Vocalization; Proceedings of the 9th International Conference on Multimodal Interfaces, pp.178-185 (2007).
- [9] Wolf, K., Schleicher, R., Kratz, S. and Rohs, M.: Tickle: A Surface-independent Interaction Technique for Grasp Interfaces, Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction, pp.185-192, (2013).
- [10] Makino, Y., Sugiura, Y., Ogata, M., Inami, M.: Tangential Force Sensing System on Forearm, Proceedings of the 4th Augmented Human International Conference, pp.29-34, (2013).