

筑波大学大学院博士課程

システム情報工学研究科修士論文

交差と囲む操作を用いたレーザーポインタ
による大画面インタラクション

久松孝臣

(コンピュータサイエンス専攻)

指導教員 田中二郎

2007年3月

概 要

近年、プロジェクタなどを用いた壁サイズディスプレイでのコンピュータ利用を想定した入力インタフェースの研究が多くなされている。本研究では、レーザーポインタを用いた壁サイズディスプレイでの入力インタフェース「リモートポインタ」を基に、従来レーザースポットを止めることによりクリックをエミュレートして GUI 操作を行っていた操作体系に替わり、動きを伴った新しい操作体系として、クロッシングや、エンサーキングなどのクリックフリーインタラクションを提案する。さらに提案を元の実装を行い、応用例としてスライドショーアプリケーションや、フォトビューワなどのアプリケーションを作成した。また、アプリケーションの試用と、実行時間や成功率（エラー率）などの基本的な性能を測定する実験を行うことにより性能評価を試みた。実験により、提案手法は研究室などの室内での利用において、画面から離れても操作性が保たれることが明らかになった。

目次

| | | |
|-------|------------------------|----|
| 第1章 | はじめに | 1 |
| 1.1 | 本研究の目的と本論文の構成 | 2 |
| 第2章 | 背景と問題点 | 3 |
| 2.1 | リモートポインタ | 3 |
| 2.2 | ホールディング手法 | 3 |
| 2.3 | ホールディング手法の問題点 | 4 |
| 第3章 | クリックフリーインタラクション | 6 |
| 3.1 | クロッシング | 6 |
| 3.2 | エンサークリング | 9 |
| 3.3 | エンサークリング&クロッシング | 10 |
| 第4章 | 実装 | 12 |
| 4.1 | レーザースポットの検出 | 12 |
| 4.1.1 | ハードウェア | 12 |
| 4.1.2 | キャリブレーション | 13 |
| 4.1.3 | レーザースポットの検出 | 14 |
| 4.2 | クロッシングの検出 | 14 |
| 4.2.1 | 当初のクロッシング検出方法 | 14 |
| 4.2.2 | クロッシング検出の改善 | 16 |
| 4.3 | エンサークリングの検出と選択判定 | 16 |
| 4.4 | アプリケーション | 20 |
| 4.4.1 | クロッシング手法を用いたアプリケーション | 20 |
| 4.4.2 | エンサークリング手法を用いたアプリケーション | 21 |
| 第5章 | 評価 | 23 |
| 5.1 | 実験1 | 23 |
| 5.1.1 | 実験方法 | 23 |
| 5.1.2 | 実験結果 | 25 |
| 5.1.3 | 考察 | 26 |
| 5.2 | 実験2 | 27 |
| 5.2.1 | 実験方法 | 27 |

| | | |
|--------------|---------------------------------|-----------|
| 5.2.2 | 実験結果 | 28 |
| 5.2.3 | 考察 | 28 |
| 第 6 章 | 関連研究 | 30 |
| 6.1 | 壁サイズディスプレイとレーザーポインタ | 30 |
| 6.2 | ペンベースシステムにおけるインタラクション | 31 |
| 6.2.1 | クロッシングに関する研究 | 31 |
| 6.2.2 | エンサークリングに関する研究 | 31 |
| 第 7 章 | まとめ | 33 |
| | 謝辞 | 34 |
| | 参考文献 | 34 |

目次

| | | |
|------|-------------------------------|----|
| 1.1 | レーザーポインタを用いた大画面インタフェース | 1 |
| 2.1 | リモートポインタのシステムイメージ図 | 3 |
| 2.2 | ホールディング手法のイメージ | 4 |
| 3.1 | クロッシング | 6 |
| 3.2 | エンサークリング | 6 |
| 3.3 | 4つの基本クロッシング | 7 |
| 3.4 | 辺とクロッシングの組み合わせに割り当てられるコマンドの例 | 8 |
| 3.5 | 辺とクロッシングの組み合わせに割り当てられるコマンドの例2 | 8 |
| 3.6 | エンサークリングによるファイル選択の例 | 9 |
| 3.7 | エンサークリング&クロッシングによるファイル選択と実行 | 10 |
| 4.1 | システムのハードウェア構成 | 12 |
| 4.2 | USB カメラで撮影した画像 | 13 |
| 4.3 | キャリブレーション | 14 |
| 4.4 | 当初のクロッシング検出方法 | 15 |
| 4.5 | ポインティングできない領域 | 15 |
| 4.6 | クロッシングの予測 | 16 |
| 4.7 | ユーザの意図しないレーザーの軌跡の交差 | 17 |
| 4.8 | レーザースポットの軌跡と記憶される座標 | 18 |
| 4.9 | 交差する線分の検出 | 18 |
| 4.10 | 記憶した座標から生成される多角形 | 18 |
| 4.11 | 多角形の面積 | 18 |
| 4.12 | 多角形の重心 | 19 |
| 4.13 | 多角形と重心の等しい正方形 | 19 |
| 4.14 | 正方形に包含される GUI オブジェクト | 19 |
| 4.15 | コマンドの割り当て | 20 |
| 4.16 | スライドめくり操作と連続操作 | 21 |
| 4.17 | フォトビューワーのスクリーンショット | 22 |
| 4.18 | サムネイルの選択 | 22 |
| 5.1 | 実験環境 | 24 |

| | | |
|-----|-----------------------------------------|----|
| 5.2 | 実験画面 | 24 |
| 5.3 | 平均実行時間 | 25 |
| 5.4 | 平均成功率 | 26 |
| 5.5 | エラーの分類 | 27 |
| 5.6 | エンサークリング手法とホールディング手法の平均所要時間比較 | 28 |
| 6.1 | Pigtail | 32 |
| 6.2 | Marking Menu | 32 |

表目次

| | |
|--------------------------------------|----|
| 3.1 辺とクロッシングの組み合わせ | 7 |
| 3.2 辺とクロッシングの組み合わせ 2 | 9 |
| 4.1 スライドショーアプリケーションのコマンド一覧 | 20 |

第1章 はじめに

近年、プロジェクタなどのような大画面と共にコンピュータを利用する機会が増大している。将来的には、画面の大型化・高解像度化によって壁サイズのディスプレイが、身近に出現することが考えられる。

出力装置であるディスプレイが大型化することに伴い、入力装置もキーボードやマウス・ペンといった従来のものから壁サイズのディスプレイに適した入力装置・入力手法が必要とされてきている。壁サイズディスプレイでのコンピュータ利用を想定した入力インタフェースの研究として、ペンを用いたもの [1]、ユーザーの身振り手振りにより入力を行うもの [2, 3, 4]、レーザーポインタを用いたもの [5, 6, 7, 8] などの研究がなされている。

著者らはレーザーポインタを用いた入力インタフェースに注目し、筆者による卒業論文 [9] に於いてレーザーポインタを用いた大画面向けポインティングインタフェース「リモートポインタ」を提案・実装した。



図 1.1: レーザーポインタを用いた大画面インタフェース

本研究では「リモートポインタ」を基盤に、レーザーポインタを用いた新たなインタラクション手法を提案する。

従来レーザーポインタを用いた研究では、レーザーポインタで「同じ」場所を照射し続けた場合クリックとみなして操作を行うといった操作体系を持つシステムが提案されている。しか

し、人間の手の震えによりレーザースポット¹が振動してしまうため、思うようにポインティングをすることができなかつたり、利用者が同じ姿勢を保とうとするため腕に力が入ってしまったりと、利用者にとって必ずしも使いやすいとは言いがたい。

本研究では、このようなレーザースポットを止めてクリック操作をする操作体系ではなく、レーザースポットを動かし続けることにより操作を行う「クリックフリー」な操作体系を提案する。一つ目はゴール・クロッシングを用いる手法、二つ目はゴール・サークリングを用いる手法である。

本研究は、2つの手法を用いることでより自然で滑らかな操作体系を持つレーザーポインタ・ベースのシステムを開発した。このシステムによって、従来のシステムでは不自然に手を止めておく必要があった操作を、静止を必要としない一連のストロークによって操作することが可能になった。本研究で提案した手法により、より使い心地の良いレーザーポインタ・ベースシステムを構築することが可能になったと考える。

1.1 本研究の目的と本論文の構成

本研究は、クロッシング手法およびエンサークリング手法を用いた、より滑らかな操作体系を持つレーザーポインタ・ベースシステムを開発し、その有効性を示すことを目的とする。

本論文ではまず第2章で筆者らがすでに開発したりモートポインタシステム [9] や、その他の従来手法の問題点を指摘し、これを解決する新たなクリックフリー手法としてクロッシング手法とエンサークリング手法を第3章に於いて提案する。さらに、第4章では第3章で提案した手法を実現するための実装について述べ、第5章に於いて実装したシステムを用いて、実行時間やエラー率（成功率）を測定することによって、提案手法の性能評価を試みる。次に第6章では、他のレーザーポインタ・ベースシステムや、提案したレーザーポインタ・ベースシステムの操作体系を考えるに当たって参考となったペン・ベースシステムのインタラクション手法などについて紹介した後、最後に第7章でまとめる。

¹レーザースポット：レーザービームがスクリーン上に照射された際に形成される光点

第2章 背景と問題点

2.1 リモートポインタ

筆者らの開発したリモートポインタシステム [9, 10, 11] では、プロジェクタによりスクリーンに投影された画面上を、レーザーポインタでレーザーを照射し動かすことでカーソルの移動を行い、画面上のクリックしたい場所を一定時間照射し続けることでクリックを行うことができる。図 2.1 にリモートポインタのシステムイメージ図を示す。リモートポインタは、プロジェクタ、PC に加え、USB 接続によるカメラを加えた構成であり、このカメラによって撮影したプロジェクタスクリーン上の画面からレーザースポットを検出して、カーソルを表示するシステムである。

リモートポインタにより、コンピュータを操作するためのポインティングデバイスとしてレーザーポインタを用いることが可能になった。

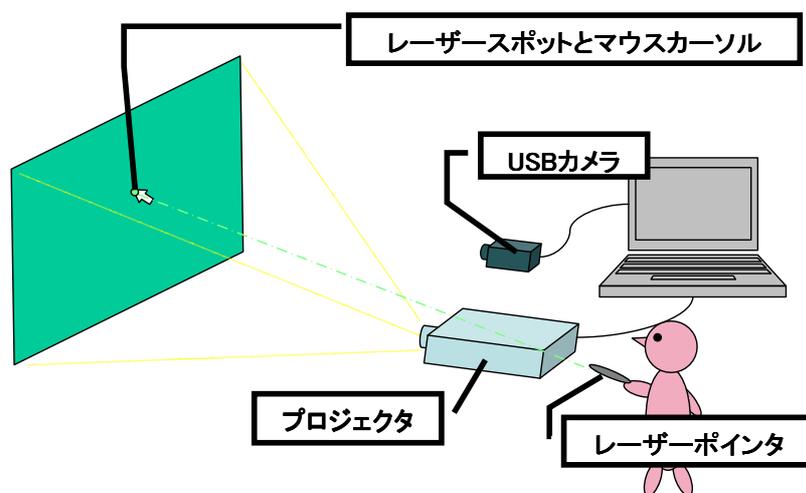


図 2.1: リモートポインタのシステムイメージ図

2.2 ホールディング手法

リモートポインタシステムでは、システムのユーザがレーザースポットを一定時間静止させておいた場合、その場所をクリックするとみなす方法をとっていた。以下これを「ホールディ

ング手法」と呼ぶことにする。図 2.2 はホールディング手法のイメージを図で表したものである。レーザーポインタをヒューマンコンピュータインタラクションに用いた研究 [5, 6, 7, 8] では、ボタンのクリックなどのインタラクションにホールディング手法と同様の手法を用いている。

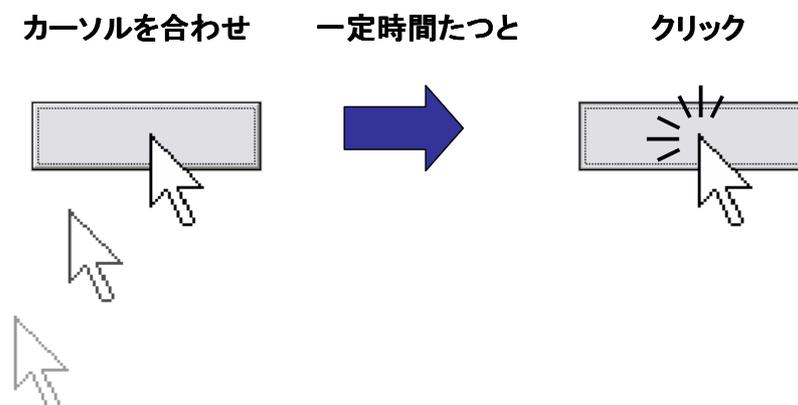


図 2.2: ホールディング手法のイメージ

2.3 ホールディング手法の問題点

Myers ら [12] は、レーザーポインタを用いたインタラクションについて、レーザーを点灯、標的を照射し、レーザーを消灯する一連の動作を行う際のレーザースポットの軌跡を解析した結果について報告している。それによると、ユーザーがレーザーポインタでターゲットを照射する場合、レーザースポットを小さな領域にとどめておくのは難しく、また、レーザーポインタのスイッチを on/off する際には、ユーザーの思い通りの場所でレーザーを点灯/消灯できないことが示されている。

筆者らのリモートポインタシステムに於いても、試用による評価によって、本システムの操作性に関する問題点が明らかになった。リモートポインタシステムでは、クリックの代替としてホールディング手法を用いているが、人間の手ブレなどにより、同じ場所をレーザーポインタで照射し続けるのは難しく、特に画面から距離をとった場合、この方法でのクリック操作は困難であることが分かった。論文 [9] では、人間の手ブレに起因するカーソルの振動を抑えるために、レーザースポットの移動距離が一定値以下である場合はレーザースポットが静止しているものとみなすことにより、カーソルの振動を軽減した。これにより、当初と比較して容易に対象をクリックすることができるようになったが、これは分解能を低下させたということであり、代償としてカーソル移動のスムーズさが損なわれ、カーソルの微調整も困難になってしまった。

また、この手法では操作に熟達した後でも、操作に時間がかかってしまう。ホールディング手法の場合、レーザースポットが既定の時間同じ位置に静止していた場合にクリックしたと

みなすため、操作速度はこの既定の時間に大きく左右されると考えられるためである。たとえば、レーザースポットを1秒静止させていた場合にクリックしたとみなすとすると、ユーザーは1秒より早くはクリックできないことになる。1秒ではなく0.5秒、0.1秒などより短い時間に設定することも可能であるが、その時間を短くするとシステムが誤認識し、ユーザーの意図しない場所・場面で不要なクリックを起こしてしまう可能性が高くなる。

ホールディング手法は、操作者に負担をかけるうえ、操作としても不自然であると考えた。本研究では、レーザースポットを静止させるホールディング手法による操作ではなく、レーザースポットを動かし、境界を横切る動作や、操作する対象を囲む動作により種々の操作を行うインタラクション手法「クリックフリーインタラクション」を提案する。

第3章 クリックフリーインタラクション

本章では、ホールディング手法にかわる新たなインタラクション手法として、クロッシング手法および、エンサークリング手法を提案する。クロッシング手法は、標的を横切ることによってコマンドを実行する手法であり、エンサークリング手法は、標的を囲むことによって操作する対象を選択する手法である。

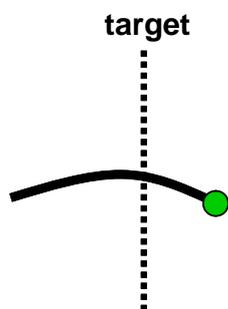


図 3.1: クロッシング

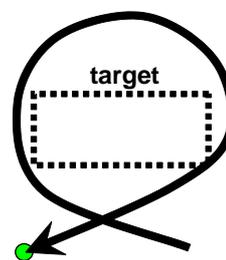


図 3.2: エンサークリング

3.1 クロッシング

クロッシング手法は、Accotらによって提唱されたゴール・クロッシング [13] を用いた、コマンド実行のための手法である。ゴールクロッシング手法は、ポインティングデバイスがボタンやメニューなどの標的（ゴール）を横切る（クロスする）することによりコマンドを実行する操作体系である。この操作体系をレーザーポインタを用いた本システムに応用する。

本システムでは、クロスするべきターゲットを表示画面の辺縁とした。画面の辺はクロスするターゲットとして十分大きく、システムの利用者にとって操作が容易であると考えられる。

本手法では、図 3.3 に示す 4 種類の基本的なクロッシングを用いることとした。すなわち次の 4 つである。

1. **In-out** 画面内から画面外へ移動するクロッシング
2. **Out-in** 画面外から画面内へ移動するクロッシング
3. **In-out-in** In-out と Out-in を連続して行うクロッシング
4. **Out-in-out** 3. とは逆に Out-in と In-out を連続して行うクロッシング

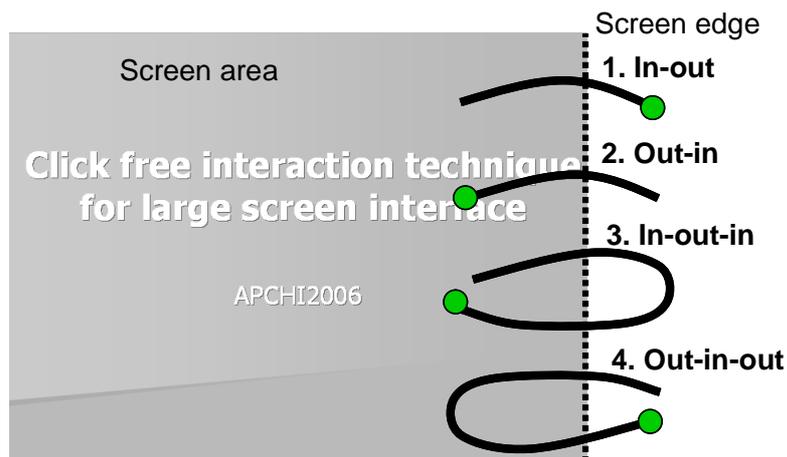


図 3.3: 4つの基本クロッシング

これらの基本クロッシングと、画面の四辺の組み合わせにそれぞれコマンドを割り当てることができる。表 3.1 には、クロッシングの種類と辺の組み合わせにより作ることのできるコマンドの一覧を示す。図 3.4 はそれらのコマンドの一例を示した図である。

表 3.1 は、組み合わせることができる一例であり、アプリケーションを実装する際には、これらの組み合わせによって作られるコマンドの中から、適宜必要なものを選択してコマンドを実装すればよい。また、Out-in クロッシングは辺に関係なくすべて同じコマンドを割り当てる、などのような実装も可能である。

表 3.1: 辺とクロッシングの組み合わせ

| 辺 | In-out | Out-in | In-out-in | Out-in-out |
|----|-----------|-----------|-----------|------------|
| 上辺 | Command A | Command E | Command I | Command M |
| 左辺 | Command B | Command F | Command J | Command N |
| 下辺 | Command C | Command G | Command K | Command O |
| 右辺 | Command D | Command H | Command L | Command P |

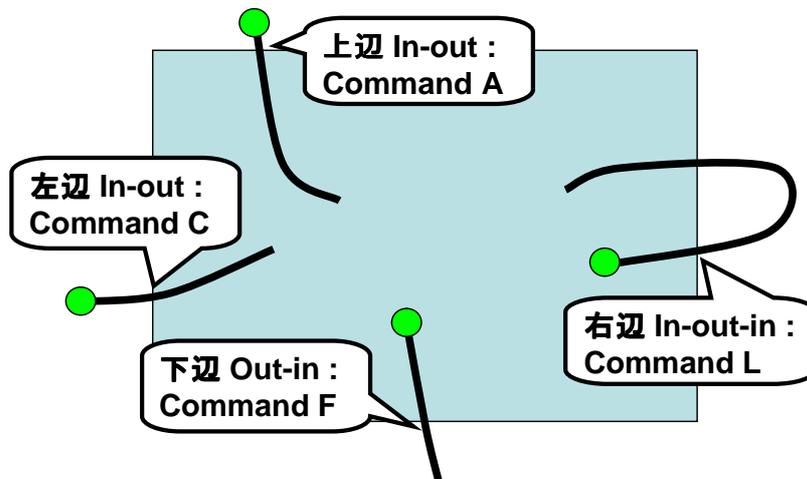


図 3.4: 辺とクロッシングの組み合わせに割り当てられるコマンドの例

さらに、In-out-in クロッシングと、Out-in-out クロッシングは、1 番目にクロスする辺と 2 番目にクロスする辺の組み合わせにより、別なコマンドを定義することもできる。たとえば、図 3.5 のように、Out-in-out クロッシングでは右辺をクロスして画面内に入り再び右辺をクロスして出て行く場合と、右辺をクロスして画面内に入り左辺をクロスして出て行く場合とで別々なコマンドを割り当てることもできる。表 3.2 は、1 番目にクロスする辺と 2 番目にクロスする辺によって別々なコマンドを割り当てる例を示した表である。

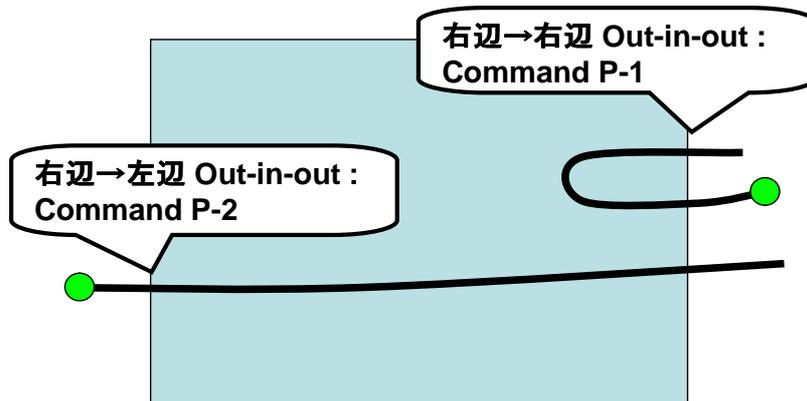


図 3.5: 辺とクロッシングの組み合わせに割り当てられるコマンドの例 2

表 3.2: 辺とクロッシングの組み合わせ 2

| | | In-out-in | Out-in-out |
|-----|-----|-------------|-------------|
| 上辺 | 上辺 | Command I-0 | Command M-0 |
| | 左辺 | Command I-1 | Command M-1 |
| | 下辺 | Command I-2 | Command M-2 |
| | 右辺 | Command I-3 | Command M-3 |
| 左辺 | 上辺 | Command J-0 | Command N-0 |
| ... | ... | ... | ... |

3.2 エンサークリング

エンサークリング手法は、GUI オブジェクトを選択するための手法である。この手法では、ユーザはアイコンやボタン、サムネイルなどの GUI オブジェクトをレーザースポットで囲むことで、オブジェクトの選択を行う。図 3.6 はファイルのアイコンを囲むことでアイコンを選択する例を示した図である。

レーザポインタを用いたシステムにおいては、円を描くような動作は比較的容易であり、本手法は、ホールディング手法によるクリック操作に比べて容易にアイコンなどのオブジェクトを選択できると考えられる。

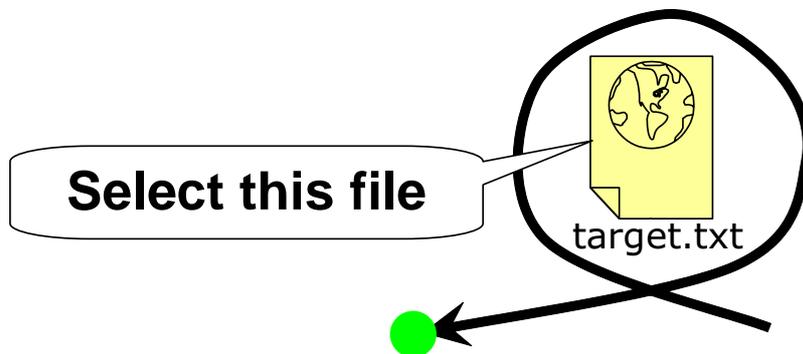


図 3.6: エンサークリングによるファイル選択の例

さらに本手法は、ユーザーがシステムに熟達した後の操作速度が、ホールディング手法に比べて早いと考えられる。ホールディング手法の場合、レーザースポットが既定の時間同じ位置に静止していた場合にクリックしたとみなすが、操作速度は既定の時間に大きく左右されることが考えられるためである。クリックとみなすのに要する時間を短くすることで操作速度を早くすることはできるが、引き換えにシステムの誤認識率が高くなると考えられる。円を描くという、特徴的な動作であれば、誤認識を少なくかつ比較的高速にターゲットの選択ができる。

また、GUI オブジェクトだけではなく、ユーザーは画面上のある領域を囲むこともできる。このことを利用して、囲んだ部分を拡大表示するといったこともできる。この場合、ユーザーは拡大したい場所と、拡大率を 1 ストロークで指定することができる。拡大したい場所は囲んだ領域の中心部分、拡大率は囲んだ面積を利用して算出すれば良い。

3.3 エンサークリング&クロッシング

3.1 節および 3.2 節では、クロッシング手法およびエンサークリング手法について説明した。クロッシング手法がコマンド実行のための手法、エンサークリング手法が GUI 選択のための手法である。これらを組み合わせることで、GUI の選択と選択した GUI へのコマンド実行を一連のストロークで行うことができる。すなわち、エンサークリング&クロッシングである。

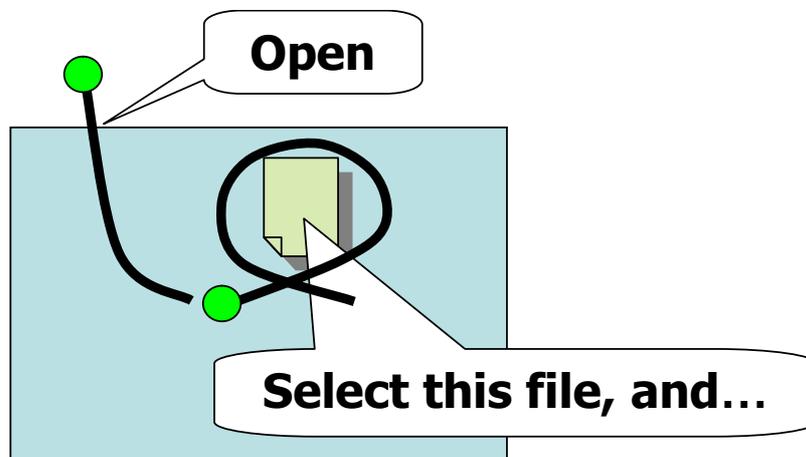


図 3.7: エンサークリング&クロッシングによるファイル選択と実行

GUI には、選択イコール実行であるものと、選択した後に何らかのコマンドを実行するものがある。

前者の例としてはプルダウンメニュー、ダイアログボックス、ラジオボタン、チェックボックス、ボタンなどがある。これらは、選択肢の中から目的のコマンドを選んで選択することでそのコマンドが実行される GUI であり、これらの GUI を操作するには、エンサークリング手法を用いれば操作することができる。スクロールバーなども、選択した後上下に動かせばよいのでエンサークリング手法によって対応できる。

後者の例としては、ウィンドウ、ファイルアイコンや画像ファイルのサムネイルなどがある。これらは、選択した後に移動する、ウィンドウを閉じる、ファイルを開く、カット・コピー・削除をするなど、対象を選択した後さらにさまざまなコマンドを実行する GUI である。

このような、選択した後様々なコマンドを実行できる GUI に対しては、エンサークリング&クロッシングが有効なのではないかと考えられる。たとえば図 3.7 は、あるファイルアイコンをエンサークリングによって選択した後に、上辺をクロッシングすることでそのファイ

ルを開くという操作を、一連のストロークで行うことができる。他の辺に移動、コピー、削除などのコマンドを割り当てることで、これらのコマンドも同様の操作で行うことができる。

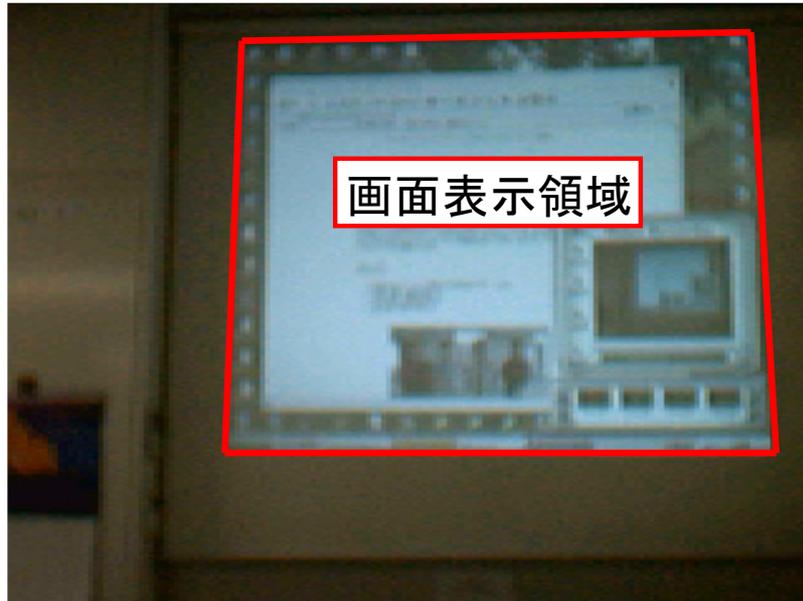


図 4.2: USB カメラで撮影した画像

4.1.2 キャリブレーション

はじめに画面の四隅を左上、左下、右下、右上の順にポインティングし、システムのキャリブレーションを行う。図 4.3 にキャリブレーション時にポインティングする箇所とその順番を示す。これらの 4 点は通常任意の四角形を構成する。これを、射影変換により表示画面のアスペクト比と同じ比を持つ長方形に変換する。射影変換を用いた手法を実装するに当たっては Sukthankar らの研究 [8] を参考にした。

射影変換を行うことで、卒業研究論文 [9] 時の実装と比較して、レーザースポットの位置検出の精度が向上した。卒業研究論文 [9] 時の実装では表示された画面および、撮影された画像内の画面が共に長方形であると仮定し、表示画面の左上、右下の 2 点をポインティングすることによりキャリブレーションを行っていた。プロジェクタおよびカメラを、スクリーンに対して真正面に配置した場合には画面のどの部分に対しても精度がほぼ一定に保たれるが、通常では、カメラをこのように厳密に配置することは難しく、カメラで撮影した画像内の表示画面は台形や互いに平行な辺のない四角形の形状を呈することが多い。そのため、画面の左上、右下付近での精度は比較的高いが、画面の左下および右上に近づく精度が低下していった。

射影変換による補正を行うことで、画面のどの位置に於いても同様の精度でレーザースポットを検出できるようになった。また、カメラを画面に対して真正面に配置する必要もなく、表示画面が収まりさえすれば斜め方向など、任意の位置に配置することもでき、設置が容易になった。

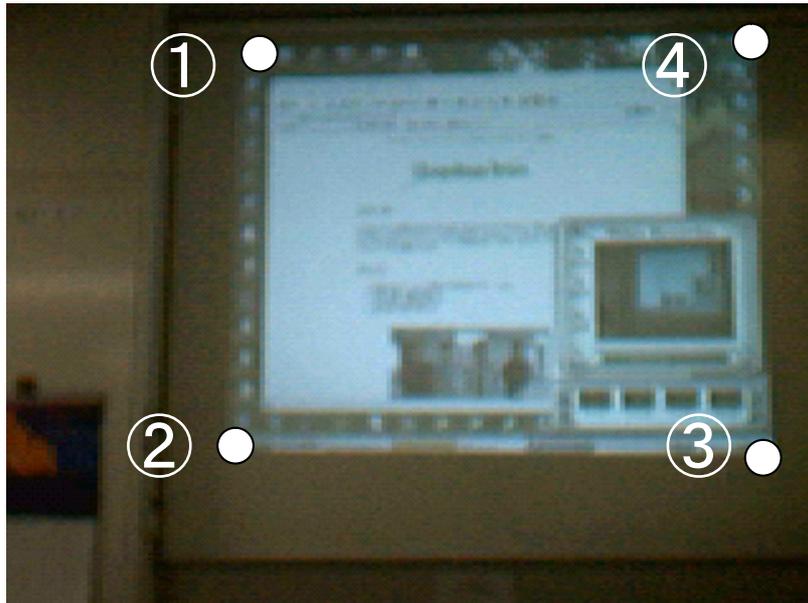


図 4.3: キャリブレーションをするポイントと順番

4.1.3 レーザースポットの検出

キャリブレーション完了後は、USB カメラから 640 × 480 ピクセルのモノクロ画像を毎秒 33.3 フレーム取得する。得られた画像をスキャンし、画像中の輝点をレーザースポットとして検出する。画像中でレーザースポットとして検出された座標に対応する画面上の座標にマウスカーソルが表示され、ユーザはマウスカーソルを介して PC を操作する。

4.2 クロッシングの検出

4.2.1 当初のクロッシング検出方法

当初の実装においては図 4.4 に示すように、USB カメラから得た連続する 2 フレーム間のレーザースポット位置を逐次比較し、一方が画面領域内に、他方が画面領域外に存在するとき画面の辺縁をクロッシングしたものと判定していた。

しかしながら、プロジェクタ用のスクリーンを用いる際には、スクリーンの内外に段差・間隙が生じることがしばしばあり、このような場合スクリーン外をポイントした際、カメラフレーム内にレーザースポットをとらえられずにクロッシングを正しく検出できない場合があることが分かった。

図 4.5 はスクリーン後方に間隙のある様子を示したものである。ユーザーから見て奥側には、スクリーンによって遮蔽されポイントできない領域が生じてしまう。ユーザーがスクリーンの横方向に立った場合などは、スクリーンに対して角度がないためにポイント

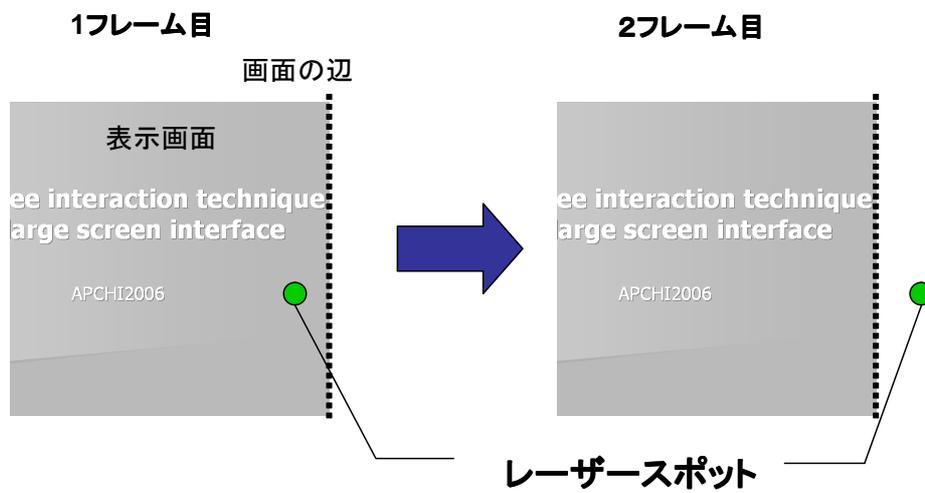


図 4.4: 当初のクロッシング検出方法

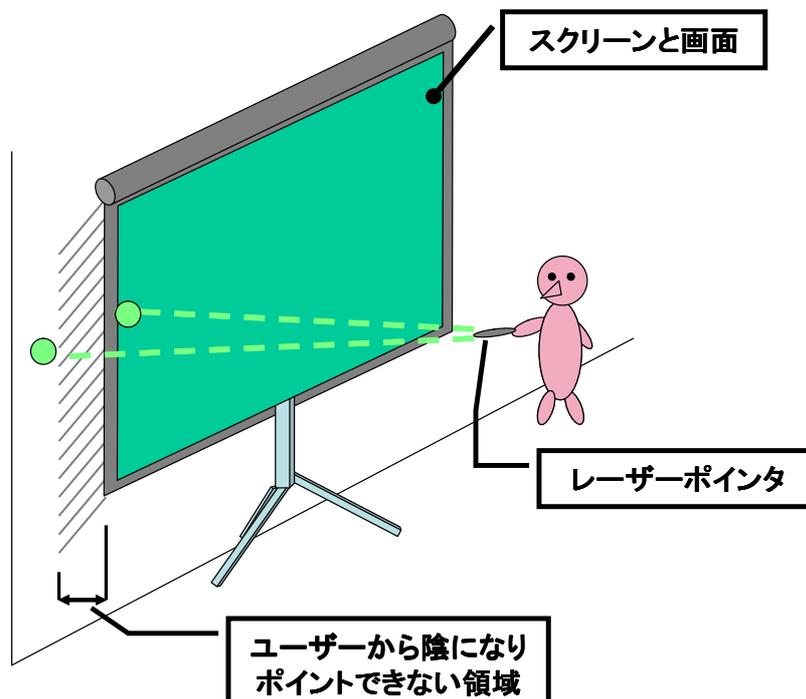


図 4.5: ポインティングできない領域

ングできない領域が大きく、スクリーン外のレーザースポットがカメラフレームに入らない場合がある。このような場合にクロッシングが検出できない。

4.2.2 クロッシング検出の改善

改善した検出方法では、スクリーン上にあるレーザースポットの位置のみを利用してクロッシングを予測する方法をとった。図 4.6 にクロッシングの予測による検出方法を図示した。図中下方の out-in クロッシングを例に説明する。検出の手順は次のようなものである。

1. 画面内に最初にレーザースポットが現れたフレームと、次のフレームのレーザースポットの位置を結んだベクトルを生成。
2. 1. のベクトルをマイナス方向に定数倍する。現在の実装では、定数 = 4 である。
3. 2. で定数倍したベクトルと画面の辺が交差する場合、クロッシングを検出。

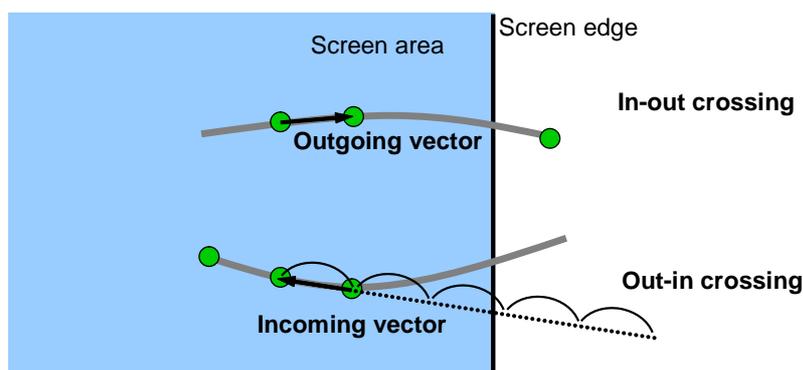


図 4.6: クロッシングの予測

また、in-out クロッシング、out-in クロッシングが一定時間内に連続して起こった場合 in-out-in クロッシングを検出する。現在の実装では、500m 秒以内に起こった場合としている。

4.3 エンサーリングの検出と選択判定

各フレームで検出したレーザースポットの位置座標を配列に格納しておく。配列には時間軸に沿ってレーザースポットが移動していく座標が格納されていく。格納された座標の前後する二つの座標から直線群を生成することができる。新たに座標が追加されるごとに、既存の座標によって構成されるそれぞれの直線と交差判定を行い、いずれかの直線と交差した場合、続けてエンサーリングの判定プロセスに入る。

エンサークリングの判定プロセスでは、まず囲まれた領域を直線群からなる多角形と近似して三角形に分割し、それぞれの三角形の面積を計算した後合計して多角形の面積を求める。この面積が一定値以上である場合に、エンサークリングが行われたと判定する。ここで、面積が一定値以上の場合としたのは、誤認識を軽減するためである。というのは、図 4.7 に示すようなユーザーの意図によらず手の震えなどからごく小さな交差が発生することがあるからである。そのため、囲まれた面積が極端に小さい場合にはエンサークリングとは判定せずに判定プロセスを終了する。

次に、囲まれた領域の中心として、多角形の重心の位置座標を計算する。そして、中心を共有する正方形をマーカーとして表示する。マーカーの正方形と多角形の面積は互いに比例するように正方形の辺の長さを決める。具体的には、多角形の面積の平方根をとり定数倍することで正方形の辺の長さを導いている。定数は、0.6 とした。この値のとき、およそレーザースポットの軌跡に内接するような正方形を表示することができる。

マーカーが、ターゲットの中心座標包含している場合、そのターゲットを選択する。

図 4.8 から 図 4.14 にエンサークリングの検出と選択判定のプロセスを図示した。

また、現在の実装では、選択するターゲットは1つのみとし、複数個のターゲットを囲んだ場合にはエラーとして何も選択しない。

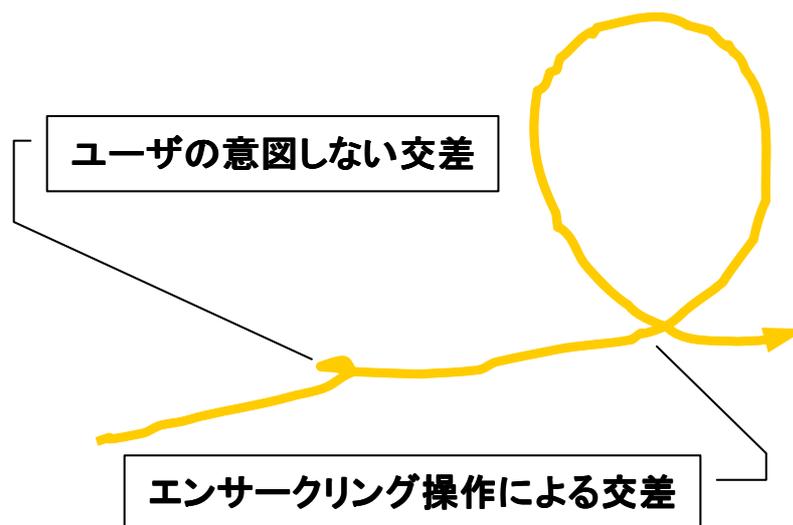


図 4.7: ユーザの意図しないレーザーの軌跡の交差

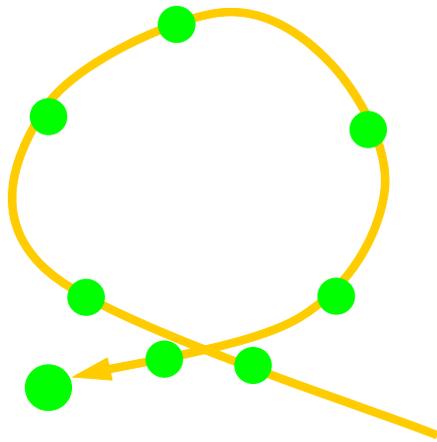


図 4.8: レーザスポットの軌跡と記憶される座標

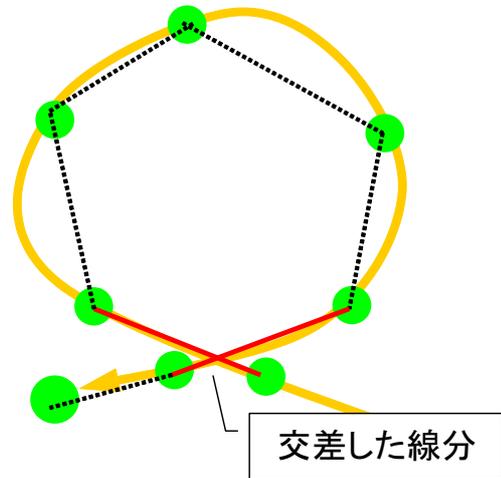


図 4.9: 交差する線分の検出

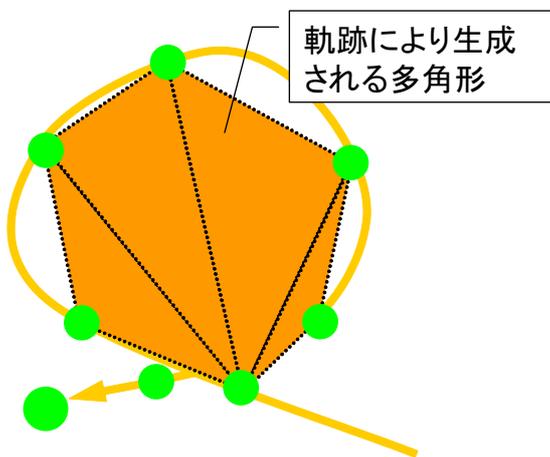


図 4.10: 記憶した座標から生成される多角形

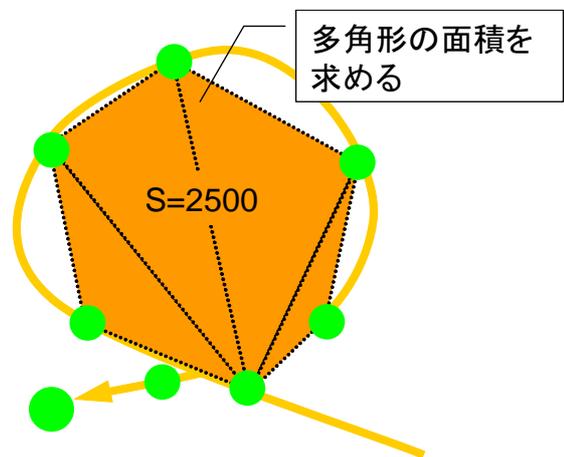


図 4.11: 多角形の面積

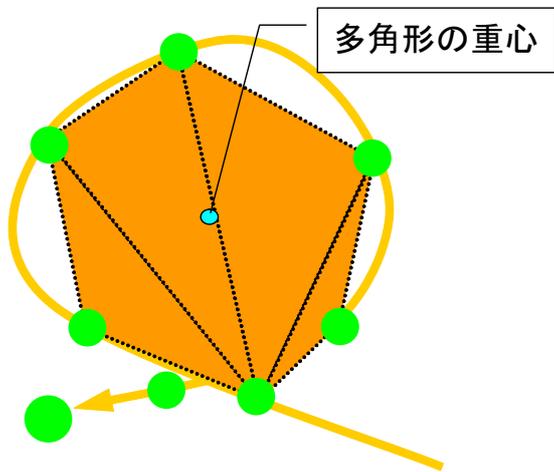


図 4.12: 多角形の重心

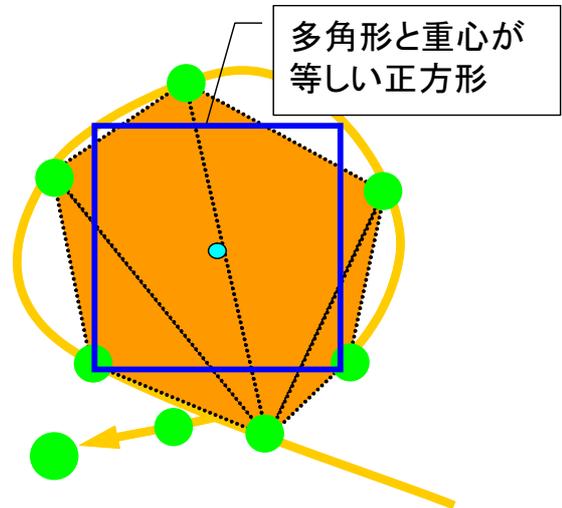


図 4.13: 多角形と重心の等しい正方形

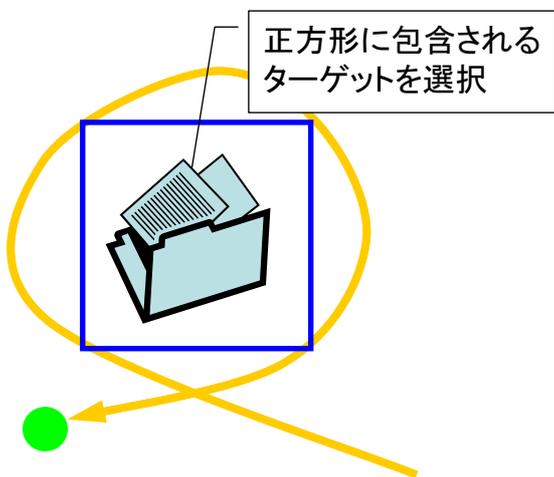


図 4.14: 正方形に包含される GUI オブジェクト

4.4 アプリケーション

4.4.1 クロッシング手法を用いたアプリケーション

スライドショーアプリケーション

クロッシング手法を用いたアプリケーションとして、スライドショーアプリケーションを作成した。このアプリケーションでは、次ページを表示、前ページを表示のページをめくるコマンドのほか、トグル操作によってスライド表示モードと、サムネイル表示モードを切り替えるコマンドなど、すべての操作をクロッシングを用いて行う。

このアプリケーションでは、基本的に Out-in クロッシングを用いて操作を行う。スライドショーを使ったプレゼンテーションでは、プレゼンテーション中にレーザーポインタでスライド内を指して聴衆に説明することがある。説明中にレーザーポインタが画面外にうっかり出てしまった場合や、うっかり外に出てしまいすぐに画面内に戻すといった動きをした場合でも誤動作をしないように、In-out クロッシングにはコマンドを割り当てず、Out-in クロッシングのみにコマンドを割り当てた。コマンドの割り当てを 図 4.15 に図示する。

表 4.1 は、クロッシングを行う辺と用いる基本クロッシングの種類とコマンドの対応関係を表で示したものである。

表 4.1: スライドショーアプリケーションのコマンド一覧

| 辺とクロッシングの組み合わせ | コマンド |
|------------------|---------------------|
| 右辺 Out-in | 次ページ |
| 左辺 Out-in | 前ページ |
| 上辺 Out-in | 上のページ (サムネイルモード時のみ) |
| 下辺 Out-in | 下のページ (サムネイルモード時のみ) |
| 一辺 他の辺 In-out-in | トグル |

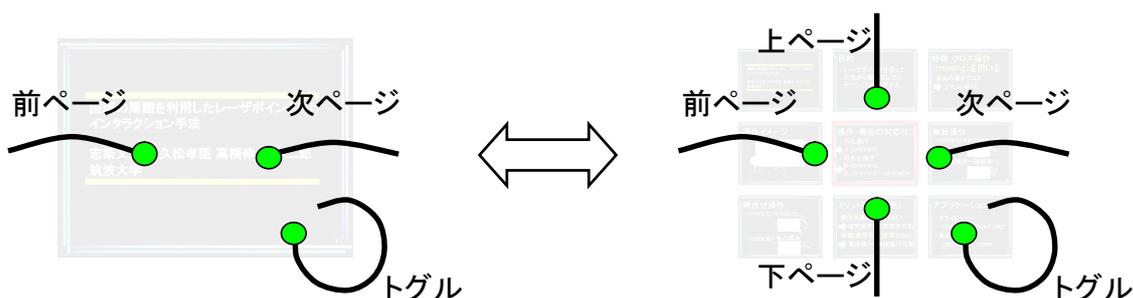


図 4.15: コマンドの割り当て

次ページを表示は右辺での Out-in クロッシング、前ページを表示は左辺で Out-in クロッシングをすることにより実行する。この際に、スライドが右や左からスライドインするアニメー

ションをすることで、次ページや前ページが表示されたのだということを示すフィードバックになっている。また、同じ辺を連続して Out-in クロッシングすることで連続的にページをめくることができる。図 4.16 にクロッシングによりスライドをめくる様子と、連続してめくる様子を示す。

トグルはスライド表示モードとサムネイル表示モードを切り替える操作である。In-out クロッシングと Out-in クロッシングが別の辺でなされるタイプの In-out-in クロッシング（たとえば右辺 下辺の In-out-in クロッシング）にトグルコマンドを割り当てた。

サムネイルモード時は、表示したいスライドにカーソルが表示され、上下左右の辺での Out-in クロッシングはカーソルを移動するコマンドとなっている。上下でのクロッシングはこのサムネイルモード時のみ利用可能である。

なお、志築および筆者らによる研究 [14] においても、画面の周囲でのクロッシングを利用した研究やアプリケーションについて述べている。

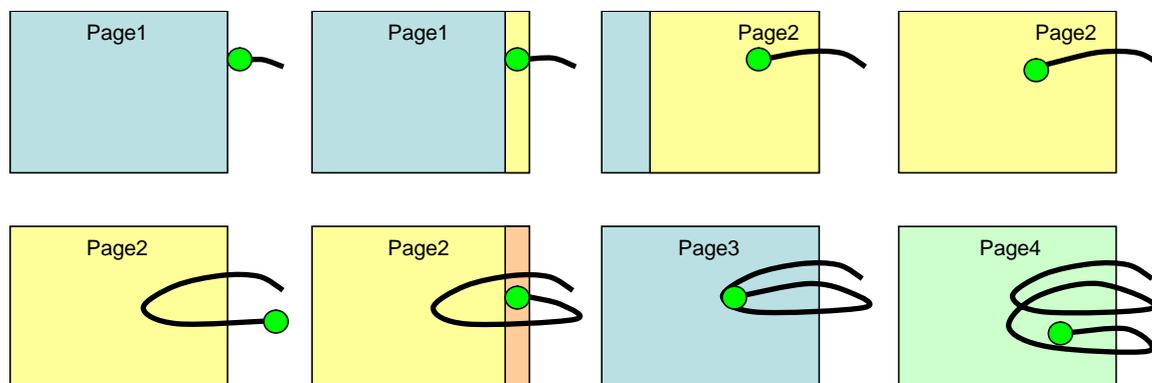


図 4.16: スライドめくり操作と連続操作

4.4.2 エンサーリング手法を用いたアプリケーション

フォトビューワー

エンサーリング手法を用いたアプリケーションとして、フォトビューワーアプリケーションを試作した。このアプリケーションでは、ウィンドウ内にサムネイル表示された画像を囲むことで、囲んだ画像を大きく表示するものである。図 4.17 にアプリケーションで写真を表示する様子を示す。レーザースポットによって囲まれた画像が中央に大きく表示されている様子が分かる。

このアプリケーションでは図 4.18 に示すように、サムネイルの中央にサムネイル本体の半分の辺の枠を想定し、この枠とエンサーリングされた際のマーカーが交差しているものを選択するように実装を行った。これは、サムネイルの端のほうを小さく囲んだ場合（図中央）や、隣りのサムネイルにまたがるように大きく囲んだ場合（図右）でも、きちんと選択できるように工夫した結果である。

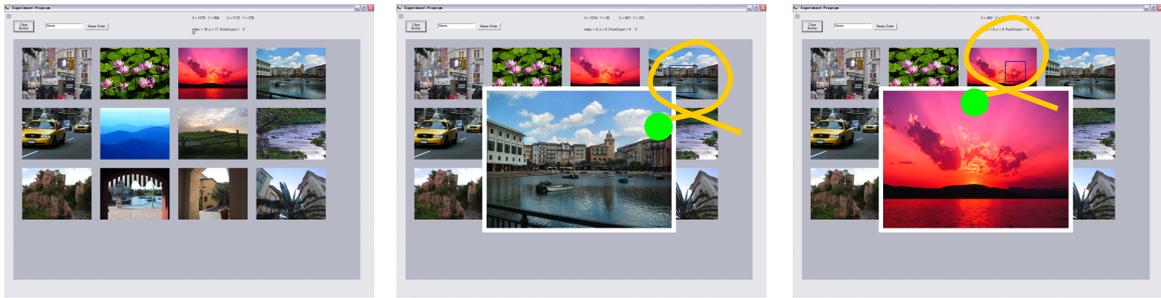


図 4.17: フォトビューワのスクリーンショット

このアプリケーションでは、サムネイルは幅 240pixel、高さ 180pixel の大きさとし、このようなサムネイルをタイル状に並べた。当初は 4.3 節で述べたように、エンサーキングによるマーカーがサムネイル中央部分を包含した場合に選択を行うこととしたのだが、この実装方法では、図左に示すような囲み方をした場合は選択できるが、図中央や、図左のように囲んだ場合どのサムネイルも選択できないことが分かり、先ほど述べたように変更した。ここで得られた知見はエンサーキング手法を用いたアプリケーションを実装する際に有用な知見である。

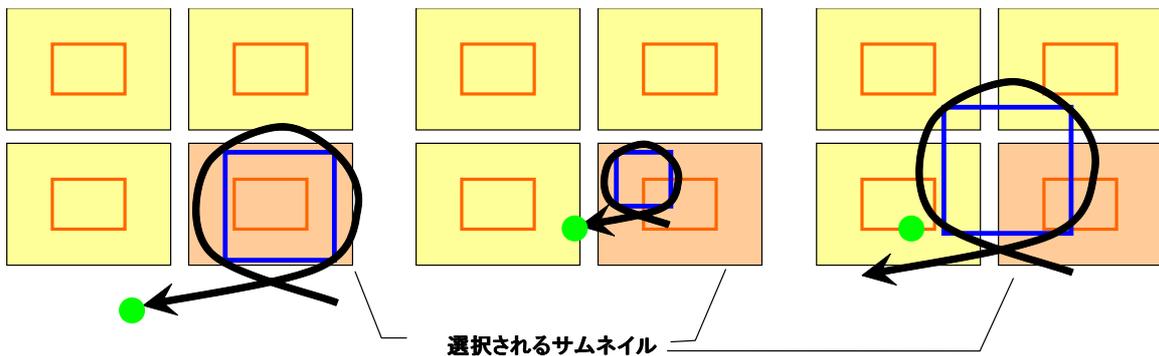


図 4.18: サムネイルの選択

第5章 評価

開発者以外の一般利用者の試用から、提案手法のうちクロッシング手法について、良好な評価が得られた。4.4.1 節で述べたスライドショーアプリケーションを実際にレーザーポインタベースのシステムを初めて使う人に利用してもらい、利用者に使用感を尋ねたところ、「操作が分かりやすい」「操作が簡単ですぐに使える」など概ね良好な感想が得られた。利用の様子を観察していても、利用者はすぐに操作になれ、特に問題なくスライド操作を行える様子が見受けられた。

さらに、エンサーキング手法について実行時間や実行成功率などの性能評価のための実験を行った。また、比較対象として、ホールディング手法・ワイヤレスマウスについても同様の実験を行った。本章では実験方法・実験結果について述べ、実験結果を元に考察を行う。

5.1 実験 1

まず、エンサーキング手法を用いた場合の実行時間および、成功率を測定した。実行時間は、被験者がひとつのターゲットを選択するのに要する平均時間を、成功率は、選択すべき全ターゲット数と被験者が実際に選択操作を行った回数の比を求めた。

5.1.1 実験方法

被験者は、プロジェクタにより投影された画面から一定の距離を保って正面に立つ。画面の解像度は 1280 × 1024 ピクセルであり、画面のサイズは、縦約 1 メートル、横約 1.5 メートルである。図 5.1 に実験環境を示す。

被験者は 21 歳から 36 歳の男性 15 名で、いずれも本研究のようなレーザーポインタベースのシステムの使用経験はない。また、被験者はいずれも腕、手指および視覚・色覚、聴覚に特に異常な点はない。

実験画面には、アイコンやサムネイルを模した正方形のマトリックスを表示した。マトリックスは縦 5、横 10 の計 50 個の正方形で構成される。正方形は一辺 90 ピクセルであり、それぞれ 10 ピクセルの間隔を取って並べられている。

被験者には、マトリックス上にひとつだけ表示される色つきのターゲットを囲むよう教示した。図 5.2 は、被験者に提示した実験画面と 1 つのターゲットを囲んで選択する例である。

被験者が色つきターゲットを囲んで正しく選択できた場合、選択成功を示すピープ音が鳴り、即座に別な位置にターゲットが表示される。被験者は続けてターゲットを囲み選択する。

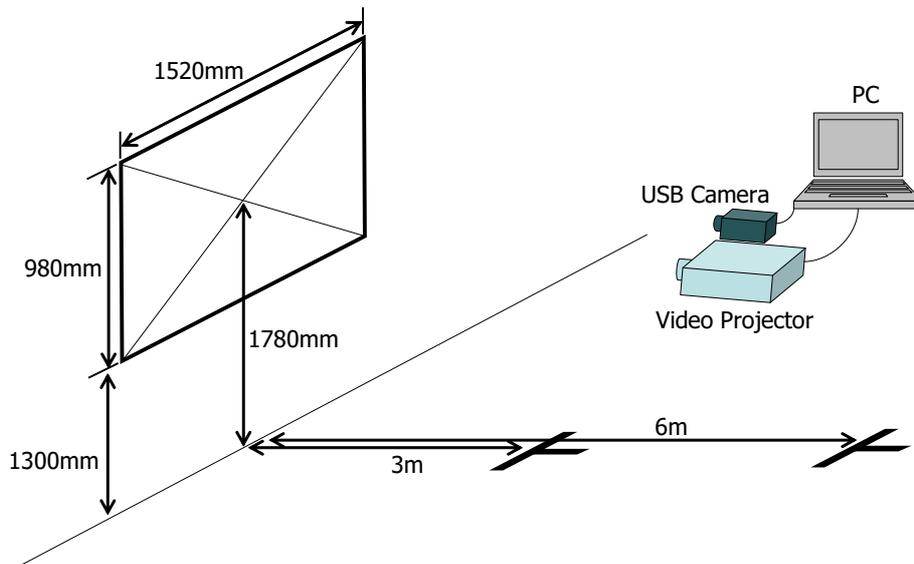


図 5.1: 実験環境

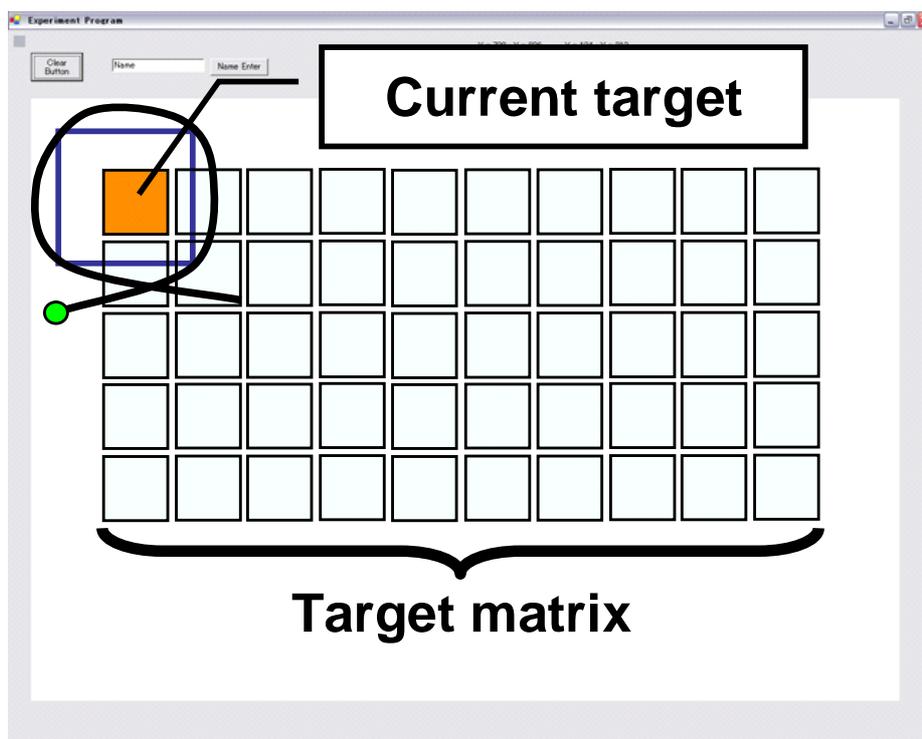


図 5.2: 実験画面

被験者が、選択に成功するまで色つきターゲットはそのまま表示される。被験者は選択に成功するまで同じターゲットを囲む。

被験者は、色つきターゲットの選択を、1セットあたり20回行い、画面から3m、6mの2地点においてそれぞれ5セットずつ行った。

5.1.2 実験結果

図 5.3、図 5.4 はそれぞれ第5セットにおける3m地点、6m地点での平均実行時間および成功率を示している。平均実行時間は囲む動作1回あたりの平均時間を示した。囲む操作1回あたりの平均実行時間は3m地点では約1.45秒、6m地点では約1.4秒であった。個人の平均では、もっとも早い被験者で平均0.9秒から1.0秒程度であった。平均成功率は、3m地点では80%~82%、6m地点では63%~78%であった。

成功率は次式によって求めた。

$$\text{成功率} = \frac{1 \text{ セットあたりの選択試行数 (20)}}{1 \text{ セットあたりの選択操作数 (20 + エラー数)}} \quad (5.1)$$

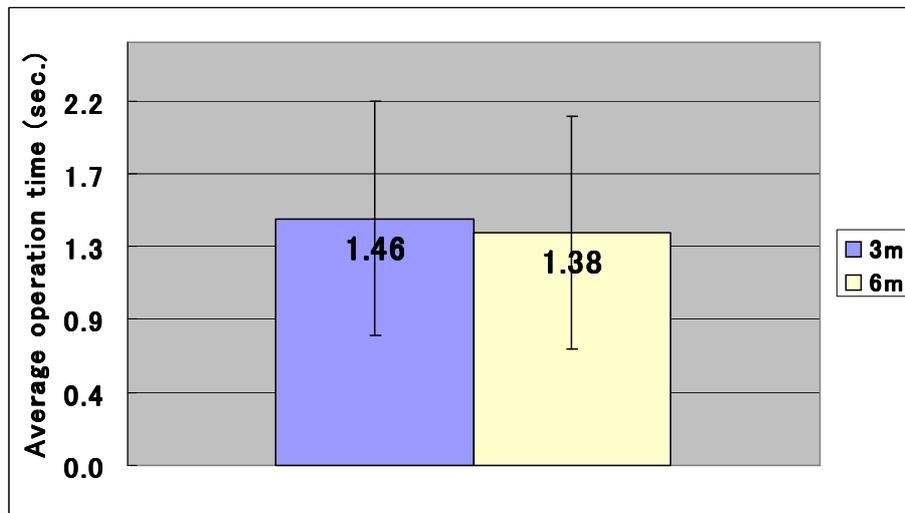


図 5.3: 平均実行時間

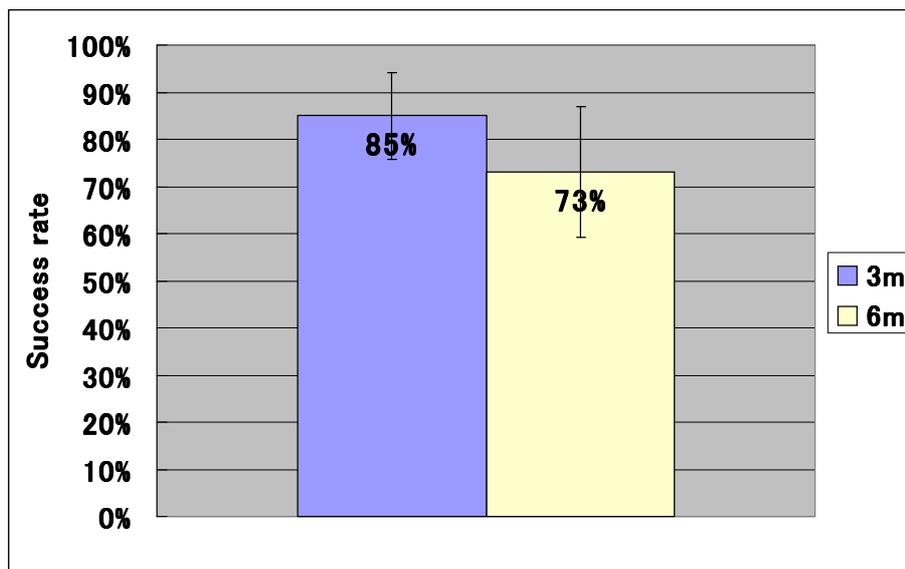


図 5.4: 平均成功率

5.1.3 考察

エラーに関する考察

実験結果より、エラーを3つのパターンに分類することができる。すなわち、

図 5.5 左 どのターゲットも選択できなかった (52%)

図 5.5 中 色つきターゲット以外のターゲットを選択した (17%)

図 5.5 右 色つきターゲットを含む複数のターゲットを選択した (31%)

の3つである。カッコ内はそれぞれのエラーの割合をパーセンテージで示したものである。1. は囲んだ領域が小さく、かつ色つきターゲットの中央を囲めなかった場合、2. は囲んだ領域の大きさは十分だが色つきターゲット以外の中央を囲んだ場合、3. は、囲んだ領域が大きすぎて色つきターゲットを含む他のターゲットを囲んでしまった場合である。

これらのエラーを削減するための方策としては次のようなものが考えられる。

1. 包含判定のアルゴリズムの変更
2. ターゲットの大きさや配置の最適化

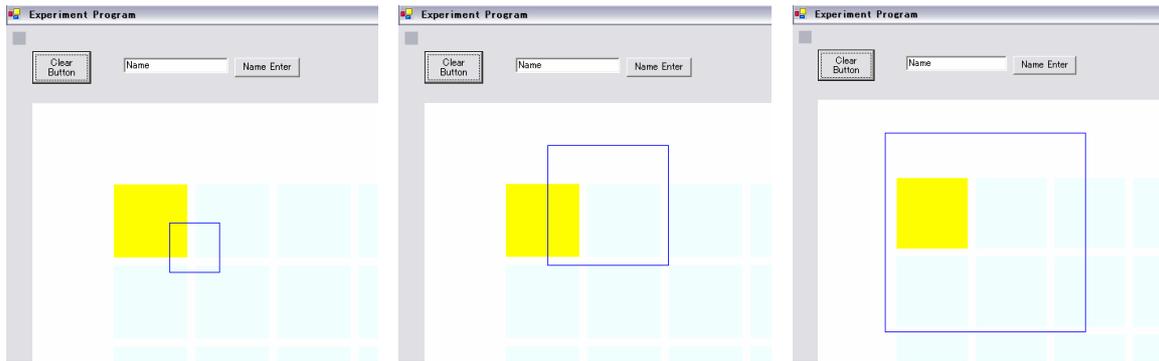


図 5.5: エラーの分類

3. 複数ターゲット選択時の絞り込み

現在、包含判定はマーカーを介して行っているが、囲んだ領域とターゲットの重なり具合によって判定を行うことなどが考えられる。また、4.4.2 節で述べたように、ターゲット中央にターゲット本体より小さい枠を想定し、枠とマーカーの交差によって判定を行う方法も考えられる。ターゲットの大きさ、配置の最適化については、ターゲットを小さくし、間隔を広くすることによって、より選択の精度が向上すると考えられる。実験では、複数のターゲットが囲まれた場合にはエラーとしたが、これをエラーとせずにユーザにさらに絞り込みを求め目的のターゲットを選択できるようにする。

5.2 実験 2

エンサークリング手法との比較として、ホールディング手法について、選択に要する時間を測定した。実験 1 および実験 2 の結果から 2 つの手法の比較を行った。

5.2.1 実験方法

実験の概要は実験 1 の方法とほぼ同様である。被験者は 5 名である。被験者には、実験 1 と同様 図 5.2 に示すような画面を提示した。

被験者には、マトリクス上にひとつだけ表示される色つきのターゲットを、ポイントし続け選択を行うよう教示した。本実験では、レーザースポットが同じ地点に 300 ミリ秒止まっていた場合、その地点にあるターゲットを選択するように設定した。ターゲットの選択、表示の仕方は実験 1 と同様次のようなものである。

被験者が色つきターゲットを囲んで正しく選択された場合、即座に別な位置にターゲットが表示される。被験者は続けてターゲットを囲み選択する。被験者が、選択に成功するまで色つきターゲットはそのまま表示される。

被験者は、色つきターゲットの選択を、1セットあたり20回行い、画面から3m、6mの2地点においてそれぞれ5セットずつ行った。

5.2.2 実験結果

実験結果を図5.6に示す。左側がエンサークルリング手法、右側がホールディング手法である。ここでは、平均時間はひとつのターゲットを選択するのに要する平均時間を示している。エンサークルリングでは、3m地点、6m地点ともに約1.8秒であり、ホールディング手法では、3m地点では、1.67秒であったのに対して、6m地点では、2.69秒とほぼ1秒の差が見られた。ホールディング手法の成功率は94%であった。

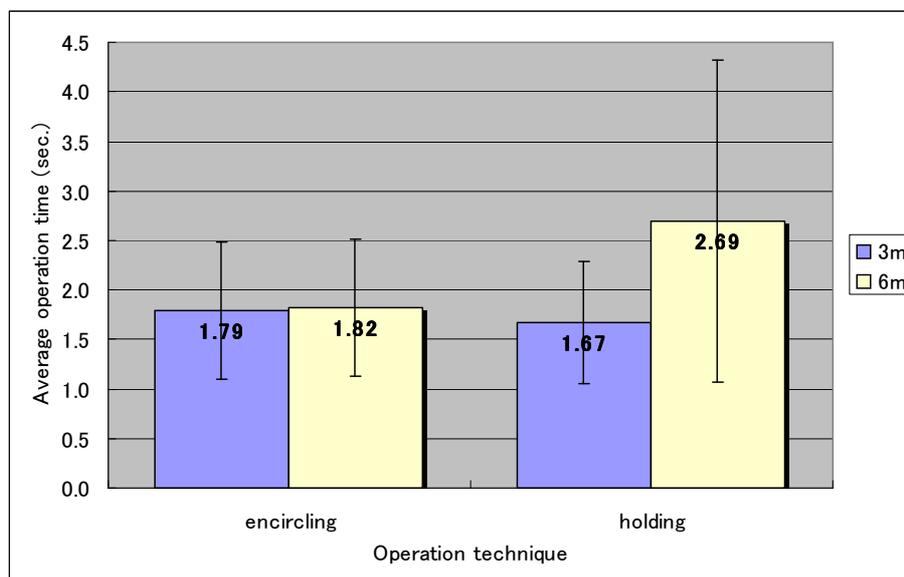


図 5.6: エンサークルリング手法とホールディング手法の平均所要時間比較

5.2.3 考察

実験結果から分かるようにエンサークルリング手法の場合、画面からの距離が3m地点と6m地点での差は見られなかったのに対し、ホールディング手法の場合、距離が遠くなるほど選択に要する時間が増大していることが分かった。このことから、ホールディング手法では距離が離れるほど操作性が低下するのに対して、エンサークルリング手法は距離が離れても操作性が低下しないといえる。これは、距離が離れるほど手元の震えが増幅され画面上でのレーザースポットの振動が大きくなることが原因であると考えられる。

ホールディング手法は、画面から近い場合には早く操作でき有用だが、画面から離れて操作する場合には不向きであり、エンサークリング手法は画面から6mまでの範囲に於いては距離にかかわらず同様の早さで操作を行うことが分かった。

第6章 関連研究

6.1 壁サイズディスプレイとレーザーポインタ

壁サイズディスプレイでの利用を想定し、レーザーポインタを入力装置として用いた研究はすでに多くなされている。Olsenら [5] や、Kirsteinら [7] は、レーザーポインタを入力装置として用いる研究を行っている。これらの研究では、ホールディング手法のようにGUIオブジェクト上でレーザースポットを留めておくことでGUIを操作したり、クリックする方法をとっている。Ohら [6] の研究では、コンピュータ制御による特別なレーザーポインタを用い、レーザーをON/OFFするボタンをさらに押し込むことでクリックなどの操作を行っている。

Sukthakarら [8] や、羽山ら [15] は、レーザーポインタを入力装置として用いたプレゼンテーション支援システムの研究を行っている。これらは、レーザーポインタで、指し示した場所を強調して表示したり、レーザーポインタにより書き込みを行ったりすることができるシステムである。これらのシステムでは、画面の隅をポイントし続けることでスライドをめくるなどの操作を行うことができるが、これらのシステムもやはりホールディング手法による操作体系であるといえる。

DavisらによるLumiPoint [16] や、田川によるAfterglow [17] などのシステムも提案されている。これらのシステムでは、レーザーポインタをペンのように用いてドローイングを行うことができるほか、プレゼンテーションを行ったり [17]、ジグソーパズルのピースを移動してパズルを組み立てたり [16] することができる。これらのシステムにおいてもアイコンにより機能を選択したり、パズルのピースを移動したりする際には、ホールディング手法と同様の操作を行っている。

Air-Real [18] は、レーザーポインタが内蔵されたりモコンを用いて電灯、エアコン、テレビ、冷蔵庫などの家電製品を操作するシステムである。このシステムの操作体系のひとつとしてPanel Operationと呼ばれる操作体系がある。エアコンを操作する際にエアコンの隣に吹き出し状の操作パネルを表示し、その内部をポイントすることで操作を行うものである。

ここで取り上げた研究はいずれもコンピュータ、プロジェクタ、デジタルビデオカメラ、といった機器で構成されており、レーザーポインタやレーザーポインタを組み込んだ装置を入力装置として用いているシステムである。これらのシステムでは、いずれもホールディング手法に類する操作体系や、特殊な装置によるボタン操作などを用いた操作体系となっており、

本研究のように画面の辺縁を横切るクロッシング手法による操作や、GUI オブジェクトなどのターゲットを囲むエンサークリング手法による操作を用いているものは見当たらない。

6.2 ペンベースシステムにおけるインタラクション

ペンベースのシステムとレーザーポインタベースのシステムは、ユーザーが操作したい箇所を直接指し示すという点で共通性があると考えられる。本研究で用いた手法は、ペンベースシステムでのインタラクション手法を参考にした部分も大きい。

6.2.1 クロッシングに関する研究

Apitz らによる CrossY [19] は、すべての操作をクロッシングによって行うクロッシングベースのドロ잉アプリケーションである。このシステムは、マウスで操作することも可能であるが、基本的にはペンで操作することを前提としたシステムである。

このシステムでは、ラジオボタン、チェックボックス、プルダウンメニュー、スクロールバー、ダイアログボックスなどに相当する GUI を、クロッシングすることによって操作を行うことができる。それぞれの GUI はこのクロッシングベースのシステムに合うようにシステム独自の形状をしている。

このシステムでは、ペンを使ってドロ잉ができるほか、ブラシ・ペンの選択、色の選択、太さの選択、スクロール操作やパレットの移動、ファイル操作などを行うことができる。

6.2.2 エンサークリングに関する研究

Hinckley らによる Scriboli [20] は Pigtail と呼ばれるテクニックを用いて Marking menu [21, 22] を表示し、表示されたメニューの中から行いたい動作を選択することによって、種々の操作を行う操作体系を持ったシステムである。Pigtail は、図 6.1 のようにペンのストロークの最後にブタのしっぽ状の形状を描くことによって操作を行うテクニックである。Scriboli では、操作を行いたいオブジェクトを囲んだ後に Pigtail を書くことによって図?? に示すような Marking menu が現れその中から実行したいコマンドを選択し実行する。

本研究では、Pigtail の手法は用いなかったものの、操作したいオブジェクトを囲むというアイデアは共通している。

Mizobuchi ら [23] は、ペン・ベースのデバイスに於いてターゲットを選択する際にペン・タッピングと囲む手法の比較を行っている。論文では、碁盤目状に並んだ正方形の中から、複数の指定されたターゲットを選択するという方式で、タッピングとサークリングの速さを測定し比較している。それによると、ペン・ベースのデバイスに於いては、概ねタッピングによるターゲット選択のほうがサークリングによるターゲットの選択よりも早い、ターゲットの並び方によってはサークリングの方が早いケースもあるという結果であった。論文では、サークリングをタッピングと併用すると効果があるのではないかと結論付けている。

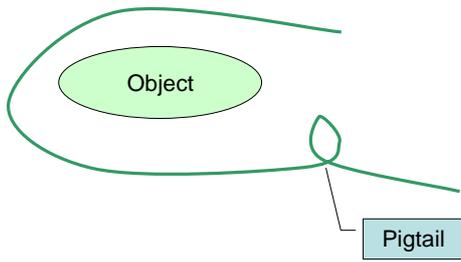


図 6.1: Pigtail

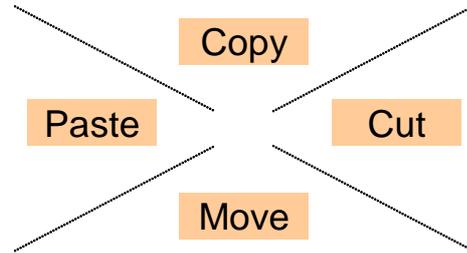


図 6.2: Marking Menu

レーザーポインタが、基本的に画面から離れて利用するのに対し、ペンは画面に接触して利用するという違いがあり、これらペン・ベースのシステムの操作体系をそのままレーザーポインタ・ベースのシステムに当てはめることはできないが、レーザーポインタ・ベースの操作体系を構築する際に参考になる研究であるといえる。

第7章 まとめ

本研究では、クロッシング手法およびエンサークリング手法を用いた、より滑らかな操作体系「クリックフリーインタラクション」によるレーザーポインタ・ベースシステムを提案・実装し、その有効性を検証するためにエンサークリング手法に関して平均実行時間や成功率などの基本的な性能を測定する実験を行った。また、従来手法としてホールディング手法についても同様の性能を測定する実験を行い、両手法の比較を行った。

実験により、画面から 3m から 6m 程度の距離では、従来のホールディング手法に対して、エンサークリング手法は画面からの距離が離れても、操作性が低下しないことが分かった。ホールディング手法は、画面から近い場合には早く操作でき有用だが、画面から離れて操作する場合には不向きであり、エンサークリング手法は画面から 6m までの範囲に於いては距離にかかわらず同様の早さで操作を行うことができることが分かった。このことから、選択操作においてはホールディング手法、エンサークリング手法を併用することで幅広い場面で利用できるシステムを作るうえで有効なのではないかと考えられる。

本研究の成果が、新しいインタラクション手法による壁サイズディスプレイの操作体系の構築に寄与することを期待する。

謝辞

本論文を執筆するに当たり、指導教員である田中二郎先生をはじめ、高橋伸先生、志築文太郎先生、三末和男先生には、丁寧なご指導と適切な助言をいただきました。心より感謝申し上げます。

また、田中研究室の皆様にもゼミや実験などを通じて大変貴重なご意見をいただきました。とりわけ、ユビキタスチームの皆様にはチームミーティングだけでなく日常的に多くのご意見やご指摘を頂きました。また岩淵志学さん、佐藤大介さんが在学中には、技術的な指導を含め貴重なご意見やご指摘を頂きました。ありがとうございました。

最後に、物心両面にわたり私を支えてくれた家族や友人にも心より感謝いたします。

参考文献

- [1] François Guimbretière, Maureen Stone, and Terry Winograd. Fluid interaction with high-resolution wall-size displays. In *UIST '01: Proceedings of the 14th annual ACM symposium on User interface software and technology*, pp. 21–30, New York, NY, USA, 2001. ACM Press.
- [2] 木村朝子, 鶴田剛史, 酒井理生, 鬼柳牧子, 柴田史久, 田村秀行. 広視野電子作業空間に関する考察とシステム試作 ~ マイノリティ・レポート型 i/f とその発展形. *インタラクシオン 2005*, pp. 143–150. 情報処理学会, 2005.
- [3] 木村朝子, 柴田史久, 鶴田剛史, 酒井理生, 鬼柳牧子, 田村秀行. ジェスチャ操作を活用する広視野電子作業空間の設計と実装. *情報処理学会論文誌*, Vol. 47, No. 4, pp. 1327–1339, 2006.
- [4] 中村卓, 高橋伸, 田中二郎. ハンドジェスチャを用いた公共大画面向けインタフェース. *マルチメディア, 分散, 協調, とモバイル (DICOMO2006)*, pp. 833–836. 情報処理学会, 2006.
- [5] Jr. Dan R. Olsen and Travis Nielsen. Laser pointer interaction. In *CHI '01: Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 17–22. ACM Press, 2001.
- [6] Ji-Young Oh and Wolfgang Stuerzlinger. Laser pointers as collaborative pointing devices. In *Graphics Interface*, 2002.
- [7] C. Kirstein. Interaction with a projection screen using a cameratracked laser pointer, 1998.
- [8] Rahul Sukthankar, Robert G. Stockton, and Matthew D. Mullin. Smarter presentations: Exploiting homography in camera-projector systems. *iccv*, Vol. 01, p. 247, 2001.
- [9] 久松孝臣. レーザーポインタを利用した大画面向けポインティングインタフェース. 卒業論文, 筑波大学第三学群情報学類, 2005.
- [10] 久松孝臣, 岩淵志学, 三末和男, 志築文太郎, 田中二郎. レーザーポインタを利用した大画面のためのインタフェース. *インタラクシオン 2005*, 2005.
- [11] 久松孝臣, 岩淵志学, 三末和男, 田中二郎. 大画面向けインタフェースへのレーザーポインタの応用. 第 19 回人工知能学会全国大会, 2005.

- [12] Brad A. Myers, Rishi Bhatnagar, Jeffrey Nichols, Choon Hong Peck, Dave Kong, Robert Miller, and A. Chris Long. Interacting at a distance: measuring the performance of laser pointers and other devices. In *CHI '02: Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 33–40, New York, NY, USA, 2002. ACM Press.
- [13] Johnny Accot and Shumin Zhai. More than dotting the i's — foundations for crossing-based interfaces. In *CHI '02: Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 73–80, New York, NY, USA, 2002. ACM Press.
- [14] Buntarou Shizuki, Takaomi Hisamatsu, Shin Takahashi, and Jiro Tanaka. Laser pointer interaction techniques using peripheral areas of screens. In *AVI '06: Proceedings of the working conference on Advanced visual interfaces*, pp. 95–98, New York, NY, USA, 2006. ACM Press.
- [15] 羽山徹彩, 国藤進. 柔軟なスライド操作を可能としたプレゼンテーション支援システム. 第 19 回人工知能学会全国大会, 2005.
- [16] J. Davis and X. Chen. Lumipoint: Multi-user laser-based interaction on large tiled displays. *Displays*, Vol. 23, No. 5, 2002.
- [17] Afterglow. URL:<<http://www.lleedd.com/afterglow/top.html>>.
- [18] 星野剛史, 堀井洋一, 丸山幸伸, 片山淳詞, 柴田吉隆, 吉丸卓志. Air-real:ホームネットワークのユーザーインタフェース. *インタラクティブシステムとソフトウェア IX*, pp. 113–118. 近代科学社, 2001.
- [19] Georg Apitz and François Guimbretière. Crossy: a crossing-based drawing application. In *UIST '04: Proceedings of the 17th annual ACM symposium on User interface software and technology*, pp. 3–12, New York, NY, USA, 2004. ACM Press.
- [20] Ken Hinckley, Patrick Baudisch, Gonzalo Ramos, and Francois Guimbretiere. Design and analysis of delimiters for selection-action pen gesture phrases in scriboli. In *CHI '05: Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 451–460, New York, NY, USA, 2005. ACM Press.
- [21] Gordon Kurtenbach and William Buxton. The limits of expert performance using hierarchic marking menus. In *CHI '93: Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 482–487. ACM Press, 1993.
- [22] Gordon P. Kurtenbach, Abigail J. Sellen, and William A.S. Buxton. An empirical evaluation of some articulatory and cognitive aspects of marking menus. In *Human-Computer Interactions*, Vol. 8, pp. 1–23. Lawrence Erlbaum Associates, 1993.
- [23] Sachi Mizobuchi and Michiaki Yasumura. Tapping vs. circling selections on pen-based devices: evidence for different performance-shaping factors. In *CHI '04: Proceedings of the*

SIGCHI conference on Human factors in computing systems, pp. 607–614, New York, NY, USA, 2004. ACM Press.