

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

動的構成が可能な
複数情報端末連携インタフェース

加茂 浩之

修士（工学）

（コンピュータサイエンス専攻）

指導教員 田中 二郎

2013年3月

概要

複数の情報端末を連携することは作業の効率を向上したり、複数人数における情報閲覧を行なったりできる点において有益である。既存の情報端末の連携手法においては(1)連携できる端末数の制限、(2)接続切り替えについての問題、(3)セキュリティの問題という3つの問題点が存在する。そこで、本研究ではこれらの問題点を解決し、かつ様々な複数の情報端末同士を動的に連携することによって、情報の閲覧と操作を円滑化するための複数画面連携インタフェースを提案した。そして、プロトタイプシステムとして Interlocked Surfaces を開発した。

プロトタイプシステムにおいて、連携できる端末数の制限についての問題は、ケーブルは用いずに無線ネットワークを利用することによって、サーバの負荷が限界に達しない限り何台でも接続することを可能にした。次に、接続切り替えについての問題については、カメラ越しに連携する対象を選択するというインタラクション手法によって直感的かつケーブルを差し替えるといった手間無く接続切り替えを実現することを可能にした。最後に、セキュリティの問題については、端末に保存された情報は一切共有せず、コンテンツと画面位置の情報のみを共有することによって、他人が端末にアクセスすることによる情報漏洩などの危険性を排除した。

さらに、プロトタイプシステムの有用性を検証するために評価実験を行なった。実験においては1人のユーザが小画面端末と大画面端末を連携することによって操作が快適になるか、情報探索の時間が短縮されるかということを検証した。その結果、システムを利用した場合、情報探索の快適さは向上したものの、探索にかかる時間は長くなった。探索時間が長くなった要因は、視線移動が発生する問題と、タイムラグの問題であると考えられる。今後は視線移動を最小限とするインタフェースの改良を検討し、タイムラグ対策としてはP2P通信を用いた実装に改善する予定である。

目次

| | |
|--|-----------|
| 第1章 序論 | 1 |
| 1.1 端末間連携 | 1 |
| 1.2 端末間連携の問題点 | 1 |
| 1.2.1 情報端末の多様化への対応 | 1 |
| 1.2.2 連携できる端末数の制限 | 2 |
| 1.2.3 接続切り替えについての問題 | 2 |
| 1.2.4 セキュリティの問題 | 2 |
| 1.3 研究の目的とアプローチ | 2 |
| 1.4 本論文の構成 | 3 |
| 第2章 システム設計 | 4 |
| 2.1 利用シーン | 4 |
| 2.2 複数画面連携インタフェースの操作 | 4 |
| 2.2.1 ペアリング対象の選択 | 4 |
| 最初のペアリングの振る舞い | 5 |
| 3台目以降のペアリングの振る舞い | 6 |
| 2.2.2 ペアリング後の画面表示 | 6 |
| 2.2.3 連動状態の制御 | 8 |
| 2.2.4 操作の連動 | 8 |
| スクロールの連動 | 8 |
| 拡大縮小の連動 | 10 |
| マウскарソル位置の共有 | 10 |
| 2.2.5 クロスプラットフォーム対策 | 10 |
| 第3章 プロトタイプの実装: Interlocked Surfaces | 11 |
| 3.1 実装方針 | 11 |
| 3.2 システム構成 | 12 |
| 3.3 クライアントサイド ペアリング部 | 13 |
| 3.3.1 カメラ搭載端末の場合 | 13 |
| 選択候補情報の重畳表示 | 15 |
| 対象選択の認識 | 15 |
| 3.3.2 カメラ非搭載端末の場合 | 16 |

| | | |
|--------------|--|-----------|
| 3.3.3 | クライアントサイド 画面連携部への遷移 | 17 |
| 3.4 | クライアントサイド 画面連携部 | 18 |
| 3.4.1 | 画面表示 | 18 |
| 3.4.2 | 操作の連携 | 19 |
| | マウスカーソルの位置情報 | 19 |
| | 現在閲覧しているコンテンツの URL 情報 | 20 |
| | スクロール位置情報 | 20 |
| | ウィンドウサイズ情報 | 21 |
| 3.5 | サーバサイド 通信部 | 22 |
| 第 4 章 | 利用シナリオ | 23 |
| 4.1 | 一人での利用例 | 23 |
| 4.1.1 | 小画面と大画面を連携して表示領域を拡張する利用例 | 23 |
| 4.1.2 | 手元の画面と遠隔の画面を連携して遠隔操作を実現する利用例 | 23 |
| 4.2 | 複数人数での利用例 | 23 |
| 4.2.1 | 複数の画面を連携して共同作業を行う利用例 | 23 |
| 4.2.2 | 遠隔地と画面を連携して共同作業を行う利用例 | 25 |
| 第 5 章 | 評価 | 26 |
| 5.1 | 実験目的 | 26 |
| 5.2 | 被験者 | 26 |
| 5.3 | 実験内容 | 26 |
| 5.3.1 | 条件 A: タブレットのみ | 27 |
| 5.3.2 | 条件 B: タブレットと画面を連携 | 27 |
| 5.4 | 実験結果 | 27 |
| 5.5 | 考察 | 35 |
| 5.5.1 | 視線移動の問題 | 35 |
| 5.5.2 | システムにおける遅延の問題 | 35 |
| 第 6 章 | 関連研究 | 37 |
| 6.1 | 複数の端末を連携して1つの画面として活用する研究 | 37 |
| 6.2 | 複数の端末を連携してそれぞれ独立して活用する研究 | 37 |
| 6.3 | 複数の端末を連携するインタラクションについての研究 | 38 |
| 第 7 章 | 議論 | 39 |
| 第 8 章 | 結論 | 40 |
| | 謝辞 | 41 |
| | 参考文献 | 42 |

目次

| | | |
|------|----------------------------------|----|
| 2.1 | カメラ映像を用いたペアリング | 5 |
| 2.2 | 最初のペアリングの振る舞い | 5 |
| 2.3 | 3台目以降のペアリングの振る舞い | 6 |
| 2.4 | 画面サイズと解像度に応じた画面表示 | 6 |
| 2.5 | コンテンツとフレームの表示 | 7 |
| 2.6 | フレーム | 7 |
| 2.7 | フレーム同士の重なり方 | 9 |
| 2.8 | スクロール連動 | 9 |
| 2.9 | フレーム移動 | 10 |
| 2.10 | フレーム拡大・縮小 | 10 |
| 3.1 | 画面同士の接続方式 | 12 |
| 3.2 | クライアントサイド ペアリング部の処理の流れ(カメラ搭載端末) | 13 |
| 3.3 | 利用可能端末のオーバーレイ表示 | 14 |
| 3.4 | 識別 ID を含んだマーカ | 14 |
| 3.5 | タッチ点の座標変換 | 16 |
| 3.6 | 点と四角形の内包判定 | 16 |
| 3.7 | クライアントサイド ペアリング部の処理の流れ(カメラ非搭載端末) | 17 |
| 3.8 | 利用可能端末のリスト表示 | 17 |
| 3.9 | 名前を入力を求めるダイアログ | 19 |
| 3.10 | 他のユーザ情報の表示 | 20 |
| 3.11 | 画面上のレイヤー | 21 |
| 3.12 | イベントの送受信フロー | 21 |
| 4.1 | 小画面と大画面を連携して表示領域を拡張する利用例 | 24 |
| 4.2 | 手元の画面と遠隔の画面を連携して遠隔操作を実現する利用例 | 24 |
| 4.3 | 複数の画面を連携して共同作業を行う利用例 | 25 |
| 5.1 | 升目状に配置されたアルファベットの組み合わせ | 28 |
| 5.2 | タスクの流れ | 29 |
| 5.3 | 実験に利用した端末 | 29 |
| 5.4 | 条件 A においてタスクを遂行している様子 | 30 |

| | | |
|-----|---|----|
| 5.5 | 条件 B においてタスクを遂行している様子 | 30 |
| 5.6 | アンケートの結果 | 31 |
| 5.7 | 3.5 インチタブレット利用時のターゲットを探し出すまでにかかった時間 . . | 32 |
| 5.8 | 7.0 インチタブレット利用時のターゲットを探し出すまでにかかった時間 . . | 32 |

第1章 序論

1.1 端末間連携

情報端末同士を連携させることは、個人か複数人を問わず作業を効果的に行なったり、遠隔操作を行なったりするために有益である。例えば、個人のデスクトップ PC とディスプレイをケーブルを用いて接続することによって、ユーザは広い作業領域で作業を行うことができ、情報の閲覧と編集を同時に行なうといった作業をより円滑に行なうことができる。さらに、スクロールやウィンドウの切り替え頻度が減少することから、作業の効率が向上すると考えられる。また、会議室において個人のノート PC とプロジェクタをケーブルを用いて接続することによって、画面の表示内容をその場にいる不特定多数の人が閲覧することができ、情報共有のために有効であると考えられる。さらに、個人のスマートフォンと自宅の PC に TeamViewer¹等のリモートデスクトップソフトウェアをインストールしておくことによって、自宅の PC を遠隔地から操作することが可能となる。

一方、我々の身の回りには数多くの情報端末が存在し、近年においてはスマートフォンやタブレット PC の普及により情報端末のスタイルが多様化している。そのため、端末の大きさと用途は多岐にわたる。例えば、スマートフォンのように小画面の端末があれば、会議室に設置されているプラズマディスプレイやプロジェクタのように大画面の端末も存在する。また、小型の端末は個人が持ち歩いて利用することに適しており、大型の端末は複数人数で情報を閲覧する際に有効である。

1.2 端末間連携の問題点

端末間連携を行なうための手段として、現在はケーブルで接続するか、リモートデスクトップソフトウェアを利用することが一般的である。しかし、これらには以下の問題点がある。

1.2.1 情報端末の多様化への対応

リモートデスクトップソフトウェアにおいては情報端末の多様化に対応し、様々な情報端末で利用可能であるものも存在する。しかし、ケーブルを用いて端末同士を接続する場合、利用可能な情報端末の種類は限られている。端末同士をケーブルを用いて接続する場合、DVI 等の規格のケーブルを物理的に情報端末に接続するが、多くの携帯端末はこれらの端子を備

¹<http://www.teamviewer.com/ja/index.aspx>

えていない。そのため、利用できる情報端末が制限されてしまっている。さらに、接続した端末間において同一の内容を画面に表示するミラーリングの場合、画面解像度が低い方に統一されてしまうため、端末のパフォーマンスを最大限に発揮することができない場合がある。

1.2.2 連携できる端末数の制限

ケーブルを用いて接続する場合、連携可能な情報端末の台数は2台に制限されてしまう。一方、リモートデスクトップにおいては1台のサーバ端末に対して複数のクライアント端末を接続することができ、3台以上の接続が可能である。しかし、サーバマシンを複数人数で操作することになるため、複数のクライアント端末において同時に操作するような状況においては操作が干渉してしまうため不適切である。例えば、リモートデスクトップを用いてクライアントAとクライアントBが文書編集を行う場合、クライアントAがページ上部を編集し、クライアントBがページ下部を編集するといったことができない。つまり、複数の端末で情報閲覧を行なうことはできるが、複数の端末から同時に情報編集を行なうことはできない。

1.2.3 接続切り替えについての問題

ケーブルを用いて接続する場合、接続する情報端末を切り替える際にケーブルを抜き差ししなければならない。例えば、会議室においてプラズマディスプレイとノートPCが接続されている場合、他のノートPCとの接続を頻繁に切り替えることは煩わしい作業であり会議の進行を妨げる可能性がある。また、ケーブルの長さの制約によって、会議室にいる全ての参加者がプラズマディスプレイを利用できない可能性がある。リモートデスクトップの場合、接続の度にIDとパスワードの入力が求められるため、動的に連携対象を切り替えるということが困難である。

1.2.4 セキュリティの問題

リモートデスクトップの場合、クライアント端末はサーバ端末の操作することができるため、悪意のあるユーザがサーバ端末の設定を変更したり、セキュリティ情報を盗み取ったりできる危険性がある。

1.3 研究の目的とアプローチ

本研究の目的は前述した(1)情報端末の多様化への対応、(2)連携できる端末数の制限、(3)接続切り替えについての問題、(4)セキュリティの問題という4つの問題点を解決し、かつ様々な複数の情報端末同士を動的に連携することによって、情報の閲覧と操作を円滑化することである。これらの問題点を解決するためには以下のアプローチを採る。

まず、情報端末の多様化への対応としては、多くのオペレーティングシステム上で動くシステムとし、異なる画面解像度の端末を連携しても端末のパフォーマンスが損なわれない設計にする。

次に、連携できる端末数については、3台以上の端末も接続可能とし、さらに操作を連携している端末全てにおいて共有することによって同時操作を実現する。

次に、接続切り替えについての問題については、物理的なケーブルは用いず、無線ネットワークを用いることによって解決する。

最後に、セキュリティ問題については、端末の全てを共有するのではなく、特定のコンテンツと操作情報のみを共有することによって解決する。

1.4 本論文の構成

本論文は、始めに第2章にてシステムの設計について述べ、第3章ではプロトタイプシステムの実装について述べる。次に第4章では提案システムの利用シナリオを示す。第5章においてはプロトタイプシステムの評価について述べ、第6章においては関連研究についての分析を行なう。次に第7章で議論を行ない、最後に第8章にて結論を述べる。

第2章 システム設計

2.1 利用シーン

我々の身の回りには数多くの端末が存在し、端末の画面の大きさや設置場所は様々である。例えば画面の大きさにおいては、携帯情報端末に搭載されている小型のものから、会議室に設置されている大画面ディスプレイのように大型のものまで存在する。また、端末を利用する場所においては、携帯情報端末のように持ち運んで利用することが多いものや、机上の計算機のようにプライベートな空間に存在するもの、会議室の大画面ディスプレイのように多くの人が集まる空間に存在するものがある。

したがって、本システムは多様な大きさの端末に対応できること、さまざまな場所で利用可能であることを考慮してシステムを開発する。

ただし、本研究が対象とする端末とは、利用者が画面の出力内容を入力によって制御できるもののみとする。そのため、デジタルサイネージといったの出力内容を利用者が制御できない画面については対象としない。

2.2 複数画面連携インタフェースの操作

本システムを用いて情報端末同士を連携することをペアリングと呼ぶ。ユーザはシステムの利用開始時にペアリングを行なう対象となる端末の選択を行なう。以降、操作している端末を操作端末と呼び、ペアリングする対象となる端末をペアリング端末と呼ぶ。さらに、既にペアリングが完了している端末のことをペアリング端末と呼ぶ。

2.2.1 ペアリング対象の選択

対象端末の選択方法は操作端末がカメラを搭載しているか否かによって異なる。これは、カメラ搭載端末と非搭載端末の両方において本システムを利用可能にするためである。

カメラ搭載端末においては、操作端末のカメラ映像越しに対象端末を選択する。カメラ映像にはペアリング可能な対象端末がハイライト表示され、対象端末の端末名や所有者情報が表示される(図2.1)。そして、カメラ映像中の対象端末を選択することによってペアリングが完了する。カメラ映像を通じて操作することによって、対象端末が複数存在する場合においても直感的に選択することができる。また、拡張現実感を用いたハイライト表示と情報提示はユーザが対象端末を選択する際の補助となる。この対象端末の選択手法はGaze-linkメタファ



図 2.1: カメラ映像を用いたペアリング

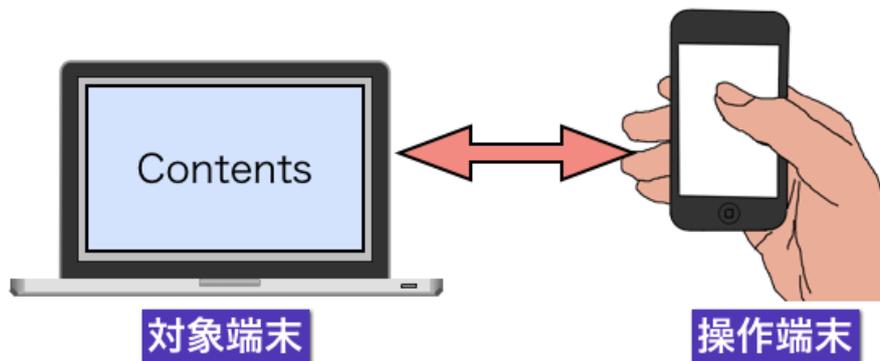


図 2.2: 最初のペアリングの振る舞い

[1][2] を応用したものである。一方、カメラ非搭載端末においては対象端末の一覧から対象を選択する。

最初のペアリングの振る舞い

ユーザが操作を行なっている端末を操作端末、ペアリングを行なう対象の端末を対象端末と呼ぶ(図 2.2)。操作端末と対象端末の両方は本システムが稼働した状態になっており、操作端末においてはペアリングを開始するモードになっている。操作端末において対象端末をペアリング対象として選択すると、対象端末の画面に表示されているコンテンツが操作端末に共有される。

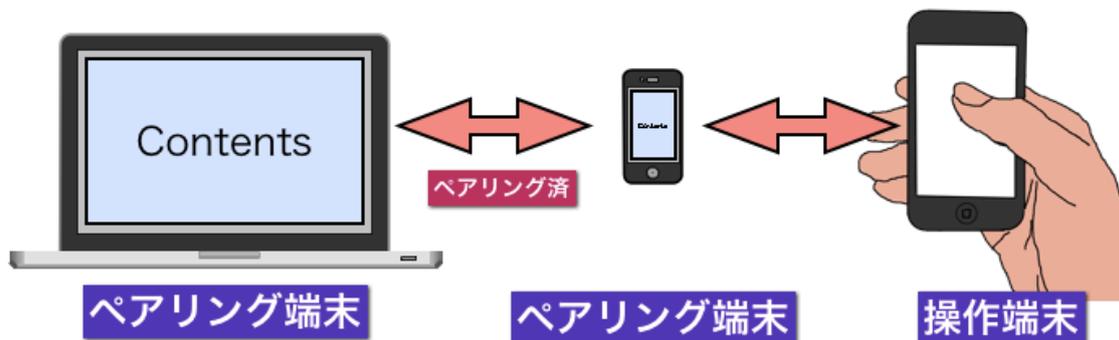


図 2.3: 3 台目以降のペアリングの振る舞い

3 台目以降のペアリングの振る舞い

既にペアリングが成立している端末をペアリング端末と呼ぶ(図 2.3)。操作端末は全てのペアリング端末を対象端末として選択することが可能である。そのため、3 台以上の端末間連携も可能である。操作端末を用いて対象端末を選択すると、既にペアリング端末の間において共有されているコンテンツが操作端末に共有される。

2.2.2 ペアリング後の画面表示

ペアリング端末にはそれぞれ画面サイズと解像度に応じた大きさのコンテンツが表示される(図 2.4)。また、別のペアリング端末の表示領域が矩形(以降フレームと呼ぶ)によって表示される(図 2.5)。フレームの内側には鍵のアイコンが表示されている(図 2.6)。これは、そのフレームが表す画面を他のペアリング端末が操作可能かどうかを表す。



図 2.4: 画面サイズと解像度に応じた画面表示

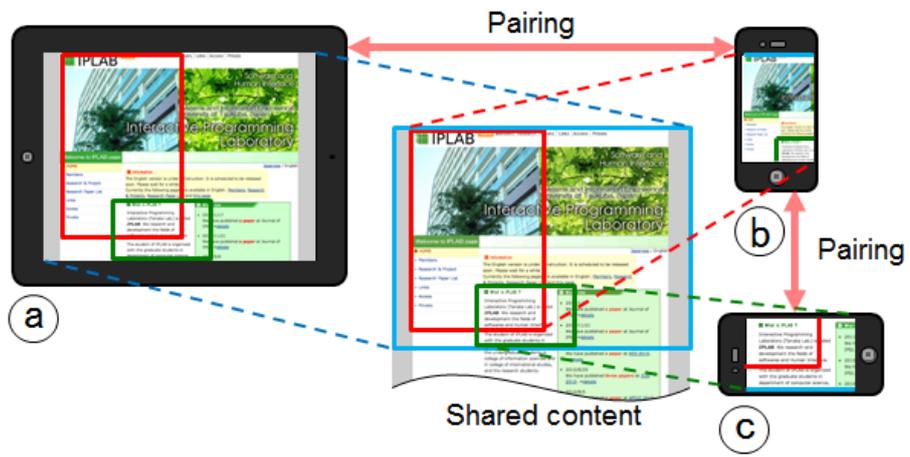


図 2.5: コンテンツとフレームの表示

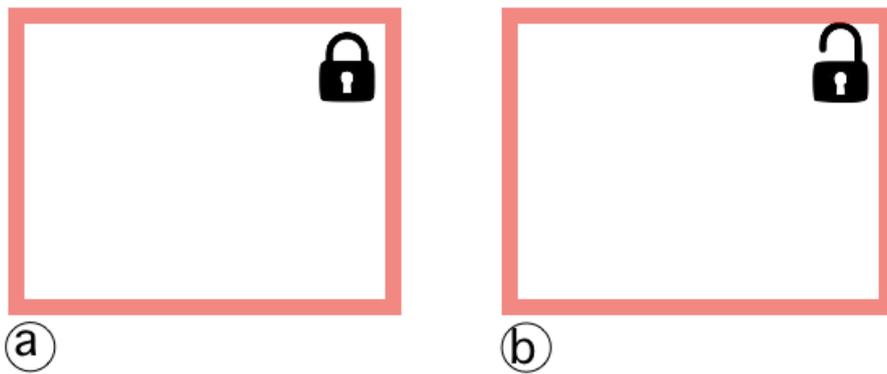


図 2.6: フレーム

2.2.3 連動状態の制御

ユーザは端末連動の状態を自由に制御することができる。これは、常に連動状態であつては不都合が発生する為である。例えば、あるユーザが自分の操作端末に文字入力をしている最中に、別のユーザからウィンドウの表示領域を変えられてしまうと作業の妨げになるといった状況が想定される。ユーザはウィンドウの右上に表示されている鍵のアイコンをクリックすることによって連動状態の制御を行なうことができる。鍵アイコンをクリックすると、施錠状態と解錠状態がトグルスイッチのように切り替わる。鍵のアイコンが施錠状態のときは、他のペアリング端末からの操作を一切受け付けない(図 2.6(a))。この状態のことを以降ロック状態と呼ぶ。一方、鍵のアイコンが解錠状態のときは、他のペアリング端末からの全ての操作を受け付ける(図 2.6(b))。この状態のことを以降アンロック状態と呼ぶ。

2.2.4 操作の連動

操作端末に対してクリックやタップ操作を行った場合、その操作は別のペアリング端末に対しても行われる。スクロールと拡大縮小操作はペアリング端末間の画面表示の違いを考慮する必要があるため、単純に操作が連動するのではなく、特殊な振る舞いをする。これらの操作は、操作端末においてはペアリング前同様に、端末独自の操作によって実現できる。さらに、ペアリング端末に対しては操作端末中のフレームを操作することによって実現できる。

スクロールの連動

操作端末に対してスクロール操作を行なう場合、この操作が他のペアリング端末に与える影響は、表示領域の重なり方によって異なる。

2つの端末の表示領域が片方によってもう片方を内包している内包状態の場合(図 2.7)、フレームは木の枠のように振る舞う。操作端末を共有コンテンツの最上部からスクロールを開始した場合(図 2.8(a))、操作端末の表示領域が別のペアリング端末のフレームの端まで到達すると(図 2.8(b))、双方が連動してスクロールするようになる(図 2.8(c))。なお、2つの端末の表示領域が重なっている交差状態の場合や、独立状態の場合は、スクロールは連動しない(図 2.7)。

ペアリング端末に対してスクロール操作を行なうには、フレームの辺をドラッグすることによって実現できる。端末 B(図 2.9(b))の表示領域が端末 A(図 2.9(a))の画面中の赤いフレームによって表されている場合、このフレームを右側にドラッグすることによって、端末 B の表示領域が右側に移動する。

なお、スクロール連動の振る舞いについて述べたが、表示領域がロック状態の場合はこの限りではない。

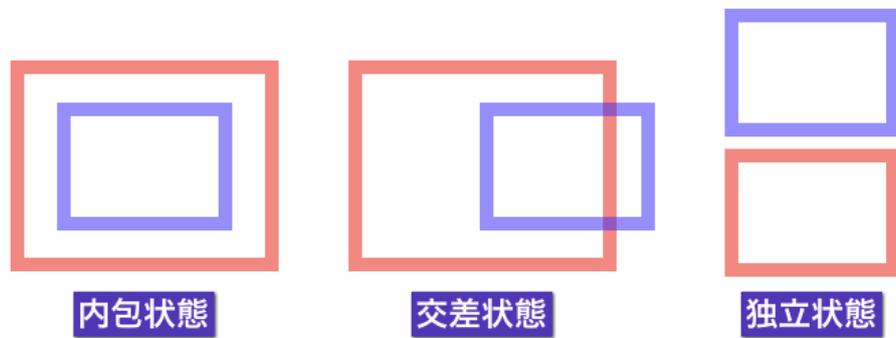


図 2.7: フレーム同士の重なり方

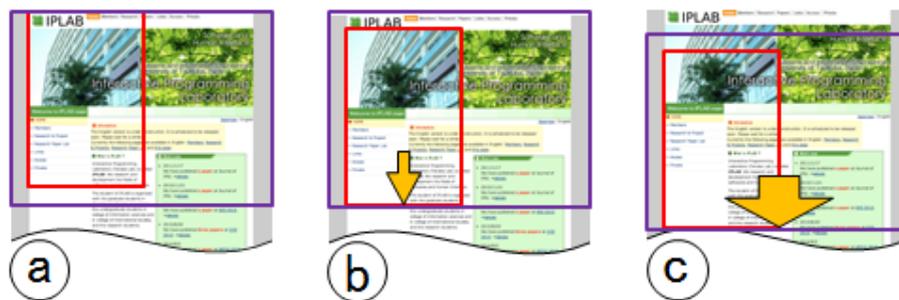


図 2.8: スクロール連動

拡大縮小の連動

操作端末に対して拡大縮小操作を行なった場合は、他の端末の表示領域には特に影響を与えない。

ペアリング端末に対して拡大縮小操作を行なう場合は、矩形の頂点をドラッグすることによって実現できる。端末 B(図 2.10(b)) の表示領域が端末 A(図 2.10(a)) の画面中の赤いフレームによって表されている場合、このフレームの角をフレームの内側に向かってドラッグすることによって、端末 B の表示領域が拡大される。なお、表示領域がロック状態の場合はこの限りではない。



図 2.9: フレーム移動

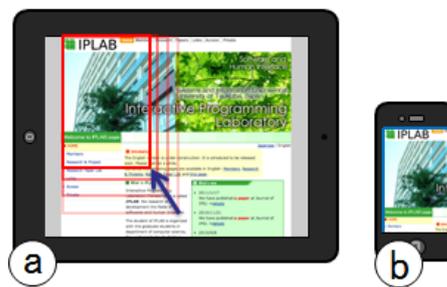


図 2.10: フレーム拡大・縮小

マウスカーソル位置の共有

操作端末に対してマウスカーソルの移動操作を行なった場合は、他の端末にマウスカーソルの位置が表示される。この機能によって、他のユーザに注目すべき場所を指し示すことが可能になる。また、他のユーザの作業状況を知ることができる。

2.2.5 クロスプラットフォーム対策

本研究が対象としている情報端末にはノート PC、デスクトップ PC、スマートフォン、タブレット端末等がある。端末によって OS や内蔵カメラの有無、画面サイズ、画面解像度などが異なるため、多くの端末において本システムを利用できるようにするためには、これらの状況を考慮する必要がある。

まず、前述したように内蔵カメラの有無については、ペアリング方法において考慮を行っている。また、画面サイズや解像度についても様々な端末に対応できる設計になっている。

次に、OS の違いについては、Web ブラウザのアプリケーションとして実現することによって対応を行っている。

第3章 プロトタイプの実装: Interlocked Surfaces

3.1 実装方針

プロトタイプシステムは Web ブラウザ Firefox¹ のアドオンとして実装した。Web ブラウザのアドオンとして実装することによって、Web ブラウザがインストールできるオペレーティングシステム上であれば環境を選ばず動作することが可能となる。論文執筆時点で Firefox は Windows と Mac, Linux, Android に対応しているため、iOS 以外のほとんどの情報端末において本システムを利用することが可能である。また、近年 Web アプリケーション上で動作するアプリケーションが多く開発されてきており、文書の作成や図の描画など、様々な作業が Web ブラウザ上で行なえるようになってきている。筆者は今後もこの流れが継続し、将来的には情報端末を利用して行なう作業のほとんどが Web ブラウザ上で行なうことができるようになる想定している。つまり、Web ブラウザのアドオンとして実装することによって、Web ページの閲覧においてだけでなく、様々な作業において利用することが可能である。

ここで、主要 Web ブラウザの各オペレーティングシステムへの対応状況を表 3.1 に示す。主要ブラウザの中でモバイル向けオペレーティングシステムである Android に対応しているのは Firefox のみである。つまり、論文執筆時点においては Firefox の Web ブラウザアドオンが最も多くの情報端末で利用可能である。

表 3.1: 主要ブラウザのオペレーティングシステムへの対応

| | Windows | Mac | Linux | Android | iOS | 備考 |
|--------------------------------|---------|-----|-------|---------|-----|-------------|
| Firefox | ○ | ○ | ○ | ○ | × | |
| Google Chrome ² | ○ | ○ | ○ | △ | △ | △: アドオン利用不可 |
| Opera ³ | ○ | ○ | ○ | △ | △ | △: アドオン利用不可 |
| Safari ⁴ | ○ | ○ | × | × | △ | △: アドオン利用不可 |
| Internet Explorer ⁵ | ○ | × | × | × | × | |

なお、画面同士を連携するための通信にはサーバとクライアントの通信を用いている (図

¹<http://www.mozilla.jp/firefox/>

²<http://www.google.co.jp/intl/ja/chrome/>

³<http://jp.opera.com/>

⁴<http://www.apple.com/jp/safari/>

⁵<http://windows.microsoft.com/ja-JP/internet-explorer/download-ie>

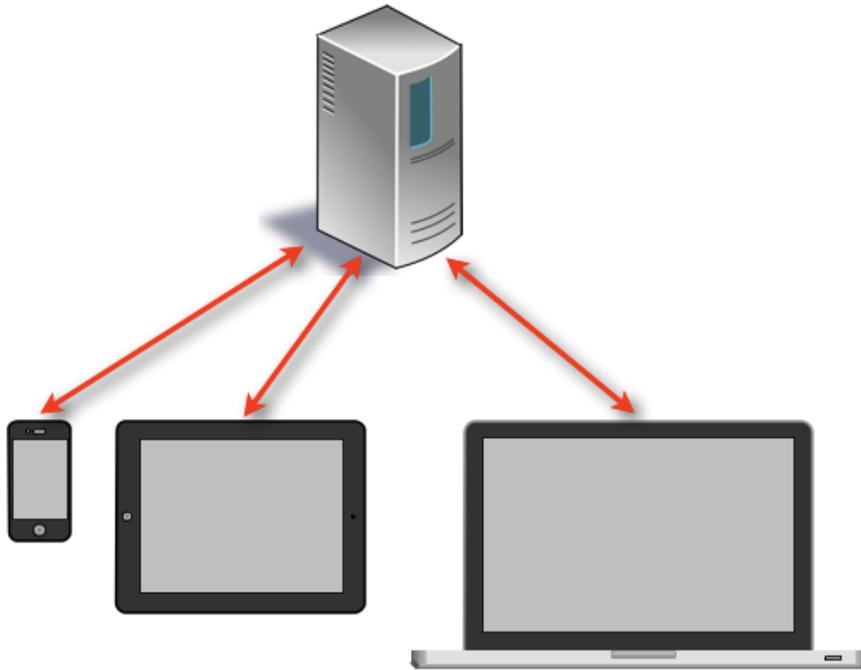


図 3.1: 画面同士の接続方式

3.1). 前述した通り、本システムは Web ブラウザのアドオンとして実装しているため、Web アプリケーションの実装と同様に JavaScript によって記述されている。したがって、サーバとの通信とは実装上の相性が良いと言える。

また、カメラ映像を用いたペアリングは Android アプリケーションとして実装した。本システムが対応しているオペレーティングシステムのうち、背面にカメラが搭載されている端末は Android のみであるためである。

3.2 システム構成

Interlocked Surfaces の主な処理は、Web ブラウザに対するユーザーの操作を他のユーザー達と共有し反映させることと、連携する対象を選択することである。

本システムは処理部を大きく3分割すると、サーバ側で処理を実行するサーバサイド通信部と、クライアント側で処理を行い、画面連携の役割を担うクライアントサイド画面連携部と、ペアリングする対象を選択する役割を担うクライアントサイドペアリング部で構成される。

3.3 クライアントサイド ペアリング部

クライアントサイド ペアリング部は Android アプリケーションとして実装した。Android 端末には背面にカメラを搭載している端末と、搭載していない端末が存在する。カメラの有無に関わらずシステムを利用可能にするため、システム側がカメラの有無によって振る舞いを変える。

3.3.1 カメラ搭載端末の場合

まず、操作端末において Android アプリケーションを起動すると、サーバにペアリング開始のリクエストを送信する (図 3.2(a))。リクエストを受け取ったサーバは、操作端末を含む全ての端末に対して利用可能端末情報を送信する (図 3.2(b))。ここでサーバから送信される情報には、利用可能な情報端末の名前とウィンドウの大きさ、識別 ID が含まれる。この情報を受け取った操作端末にはカメラ映像が表示され、操作端末をその他の端末に向けて、その端末が利用可能であることを示す枠とその端末の名前が重畳表示される (図 3.3)。また、この情報を受け取ったその他の端末は、サーバから送信された識別 ID に基づいたマーカ (図 3.4) を画面上に表示する。なお、このマーカの内部に書かれている 2 桁の数字が識別 ID である。カメラ映像中の端末を選択すると、サーバへペアリング対象の情報を送信する (図 3.2(c))。この情報を受信したサーバは、全てのクライアントに対してペアリングが確定したことを通知する (図 3.2(d))。この通知には現在端末間で共有されているコンテンツの情報が含まれる。この通知を受け取った操作端末は Web ブラウザを起動して、処理は画面連携部へ遷移する。また、この通知を受け取ったその他の端末はマーカの表示を消す処理を行なう。

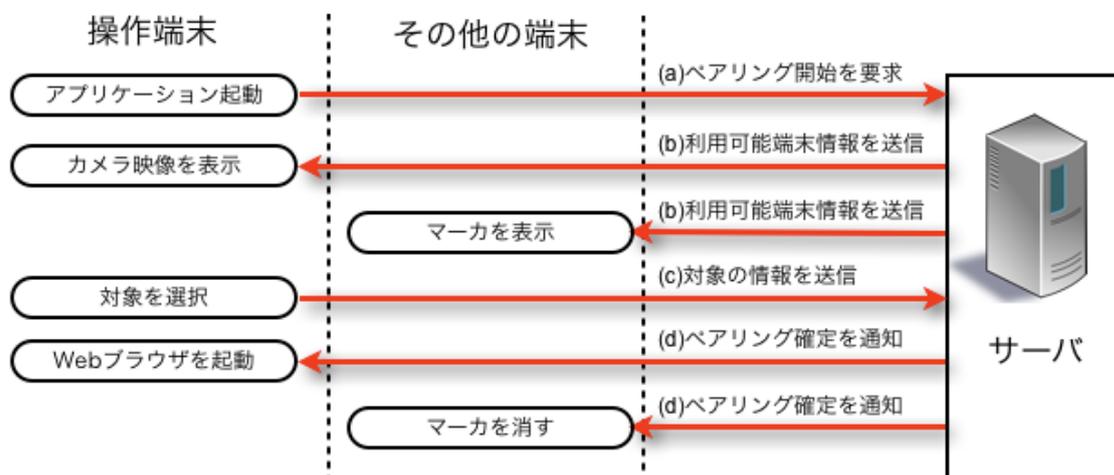


図 3.2: クライアントサイド ペアリング部の処理の流れ (カメラ搭載端末)



図 3.3: 利用可能端末のオーバーレイ表示

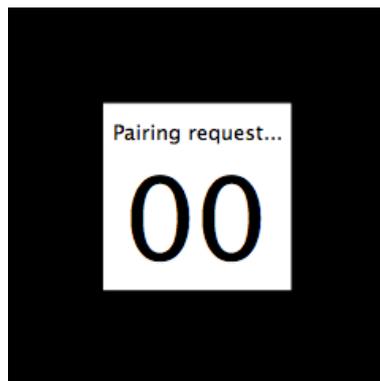


図 3.4: 識別 ID を含んだマーカ

選択候補情報の重畳表示

カメラ映像の読み込みや、カメラ映像への選択候補情報の重畳表示には AndAR ライブラリ²を利用した。AndAR ライブラリとは加藤ら [3] が開発した ARToolKit を Android 向けに移植したものであり、図 3.4 のように黒枠で囲まれた矩形を認識して、位置姿勢情報を計算することができる。また、矩形内のパターンをマッチングすることによって、マーカの個体識別を行なうことができる。

情報端末をマーカのかざすとマーカを認識して画面の個体識別を行ない、サーバから受信した端末の名前と大きさの情報に基づいて矩形の枠と名前を拡張現実感を用いて重畳表示している。

対象選択の認識

拡張現実感を用いて重畳表示されて枠の領域内に対してタッチ操作を行なうことによって、対象を選択することができる。この操作を実現するためには、タッチを行なった画面上の二次元座標から、その点が三次元で表される枠の内側かどうか判断する必要がある。三次元で表される枠の四隅の座標のスクリーン上の二次元座標は次のように求めることができる。

まず、タッチを行なった座標 (x, y) は左上を原点とし、 x 軸方向の最大値を W 、 y 軸方向の最大値を H とする座標系として求められる (3.5 左)。ただし、 W は端末固有の画面の横幅、 H は画面の高さである。この座標を画面の中心を原点とし、 x 軸と y 軸共に正方向は 1.0、負方向は -1.0 が最大となる座標系における座標 (x', y') に変換する (3.5 右)。以降、ここでの変換後の座標系をスクリーン座標系と呼ぶ。

次に、AndAR ライブラリによって、マーカの中心位置 (X_m, Y_m, Z_m) をカメラ座標系における位置 (X_c, Y_c, Z_c) に変換する座標変換行列 (式 3.1) を得ることができる。ただし、 $r_1 \sim r_9$ は回転成分を表しており、 t_x, t_y, t_z はそれぞれ並進成分を表している。マーカの中心を原点とする座標系における、フレームの四隅のうちのいずれかの点 F_m の座標を (x_{mf}, y_{mf}, z_{mf}) とすると、スクリーン座標系における点 F_s の座標 (x_{sf}, y_{sf}, z_{sf}) は座標変換行列を用いた式 3.2 で求めることができる。

以上よりスクリーン座標系におけるフレーム四隅の座標と、タッチ点の座標を求めることができたので、タッチ点がフレームの四隅の点で構成される四角形に内包されるかの判定を行なう。図 3.6 のようにタッチ点が四角形内にある場合、同色で示されている 2 つのベクトルに着目すると、いずれの場合においても全てのベクトルが反時計回りとなる。しかし、タッチ点が四角形の外側にある場合、時計回りをするベクトルの組み合わせが発生する。つまり、色分けしたベクトルの外積を計算し、求められた外積を正規化したものが全て等しい場合にタッチ点が四角形の内側にあることになる。

²<http://code.google.com/p/andar/>

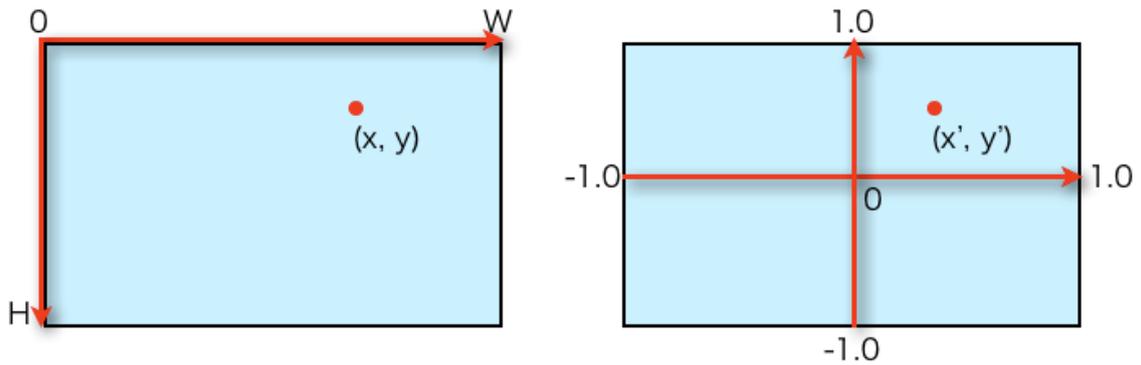


図 3.5: タッチ点の座標変換

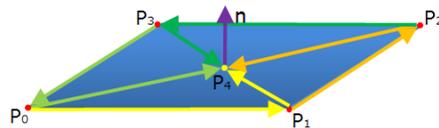


図 3.6: 点と四角形の内包判定

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} r_1 & r_2 & r_3 & t_x \\ r_4 & r_5 & r_6 & t_y \\ r_7 & r_8 & r_9 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix} \quad (3.1)$$

$$\begin{bmatrix} x_{sf} \\ y_{sf} \\ z_{sf} \\ 1 \end{bmatrix} = \begin{bmatrix} r_1 & r_2 & r_3 & t_x \\ r_4 & r_5 & r_6 & t_y \\ r_7 & r_8 & r_9 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{mf} \\ y_{mf} \\ z_{mf} \\ 1 \end{bmatrix} \quad (3.2)$$

3.3.2 カメラ非搭載端末の場合

まず、操作端末において Android アプリケーションを起動すると、サーバにペアリング開始のリクエストを送信する (図 3.7(a))。リクエストを受け取ったサーバは、操作端末を含む全ての端末に対して利用可能端末情報を送信する (図 3.7(b))。ここでサーバから送信される情報には、利用可能な情報端末の名前とウィンドウの大きさ、識別 ID が含まれる。この情報を受け

取った操作端末には利用可能な情報端末の一覧が表示される (図 3.8)。リストの中から情報端末を選択すると、サーバへペアリング対象の情報を送信する (図 3.7(c))。この情報を受信したサーバは、全てのクライアントに対してペアリングが確定したことを通知する (図 3.7(d))。この通知には現在端末間で共有されているコンテンツの情報が含まれる。この通知を受け取った操作端末は Web ブラウザを起動して、処理は画面連携部へ遷移する。



図 3.7: クライアントサイド ペアリング部の処理の流れ (カメラ非搭載端末)

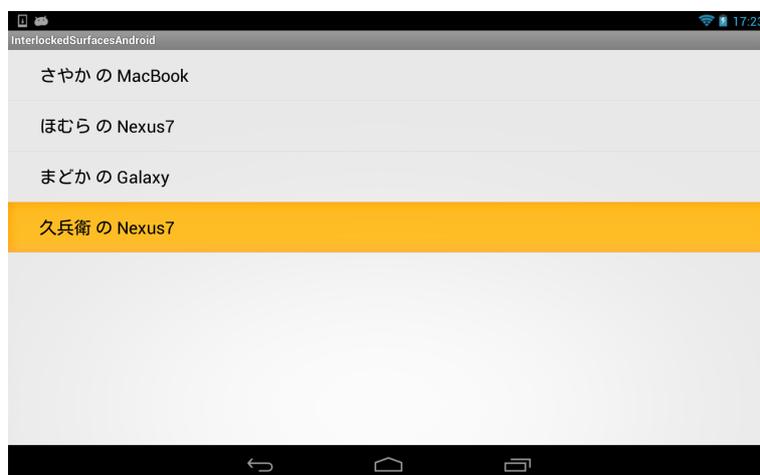


図 3.8: 利用可能端末のリスト表示

3.3.3 クライアントサイド 画面連携部への遷移

対象端末を選択すると、クライアントサイド画面連携部へ遷移する。前述した通り、画面連携部は Web ブラウザアドオンとして実装されているため、Android アプリケーションである

ペアリング部から Firefox に切り替えなければならない。アプリケーションの切り替えをユーザが行なうというのは現実的でないため、システム側で自動的にアプリケーションを切り替え、ユーザには切り替えていることを感じさせないことが望ましい。そこで、アプリケーションの切り替えには、Android のインテントという機構を用いた。ペアリング確定の通知には現在端末間で共有されているコンテンツの URL 情報が含まれている。この URL をインテントを利用して自動的に開くことによって、明示的にアプリケーションを切り替えることなく遷移することができる。

3.4 クライアントサイド 画面連携部

クライアント画面連携部は Firefox のアドオンとして実装されており、実装言語には Javascript を用いている。画面描画のためのライブラリには Raphael.js¹ を使用しており、サーバとの通信を行なうためのライブラリには node.js² を利用しており、node.js を利用したリアルタイム通信には express³ フレームワークを用いて開発を行った。本システムがインストールされた Firefox を起動すると、ユーザに名前を入力を求めるダイアログが表示される (図 3.9)。ここで入力した名前は、画面共有に参加している他のユーザに共有される。また、他のユーザの画面情報はお互いの画面に表示される。連携したユーザ同士では以下の情報が共有される。

- マウスカーソルの位置情報
- 現在閲覧しているコンテンツの URL 情報
- スクロール位置情報
- ウィンドウサイズ情報
- 連携状態の情報

3.4.1 画面表示

画面には共有している Web コンテンツに加えて、共有に参加している他のユーザのウィンドウの位置情報とユーザ名、マウスカーソル位置が表示される 3.10。Web コンテンツにシステムが出力する情報を重ねて表示をするために、Web コンテンツに同じ大きさの透明なレイヤーを重ね合わせ、このレイヤーに対して描画を行なっている (図 3.11)。このレイヤーは、Web コンテンツの DOM(Document Object Model) を解析し、要素を追加することによって実現している。システムは Web ブラウザが DOM を構築する度に解析を行ない透明レイヤーの DOM を付加している。

¹<http://raphaeljs.com/>

²<http://nodejs.jp/>

³<http://expressjs.com/>



図 3.9: 名前の入力を求めるダイアログ

しかし、透明なレイヤーを表面に表示すると、Web ブラウザに対する操作がコンテンツに反映されないという問題が発生する。Web ブラウザに対する操作とは、マウスのクリック操作やタッチパネルへのタッチ操作である。これらの操作のイベントを Web コンテンツではなく透明レイヤーが受け取ってしまうのである。つまり、Web コンテンツ上のリンクをクリックしてもその操作が反映されなくなるのである。

この問題に対応するために、透明レイヤーにユーザの操作に対するイベントハンドラを用意し、透明レイヤーにおいて発火したイベントを Web コンテンツに対して再現することによって、透明レイヤーと Web コンテンツの両方が操作を受け取ることができる。

3.4.2 操作の連携

操作端末に対して行なった操作をイベントと呼ぶ。イベントにはマウスカーソルの移動、表示しているコンテンツの URL の変更、スクロール位置の移動、ウィンドウサイズの変更があり、これらのイベントを連携している端末同士で共有している。ある端末でイベントが発生した場合、そのイベント情報をサーバに送信する(図 3.12)。そして、イベント情報を受け取ったサーバはそれ以外の端末にイベント情報を送信する(図 3.12)。他の端末のイベント情報を受け取ったクライアント端末は、そのイベント情報を画面に反映させる。次に、それぞれのイベントにおいてどのような情報をやりとりしているかについて述べる。

マウスカーソルの位置情報

操作端末におけるマウスカーソルの移動操作は Web ブラウザにおける `mousemove` イベントをフックすることによって取得している。フックしたイベントからマウスカーソルの x 座標と y 座標を取得し、この情報をサーバに送信している。



図 3.10: 他のユーザ情報の表示

このイベントを受信したペアリング端末は、受信したイベントの座標にマウスカーソルの画像を表示している。

なお、位置が表示されるのは Android 以外の端末のみである。Android はタッチによって操作するのでマウスカーソルという概念が存在しないため、カーソル位置は表示されない。

現在閲覧しているコンテンツの URL 情報

操作端末においてページの移動が行なわれたとき、Web ブラウザにおいて document オブジェクトの要素である href の値が変わる。href は現在閲覧しているページの URL 情報が格納されている。この URL 情報をサーバに送信する。

サーバからイベントを受信したペアリング端末は、同様に document オブジェクトの href 要素の値を書き換えることによってページ移動を実現している。

スクロール位置情報

操作端末におけるスクロール操作は Web ブラウザの window オブジェクトの scroll イベントをフックすることによって取得している。フックしたイベントからスクロール位置の x 座標と y 座標を取得し、この情報をサーバに送信している。



図 3.11: 画面上のレイヤー

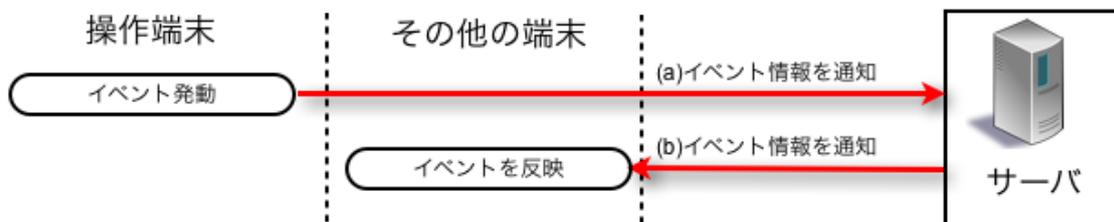


図 3.12: イベントの送受信フロー

このイベントを受信したペアリング端末は、window オブジェクトの要素である scrollX と scrollY の値を変更することによってクライアントのスクロールを実現している。

なお、フレームの辺に対してドラッグ操作が行なわれた場合は、スクロールイベントをサーバに送信することによって実現している。

ウィンドウサイズ情報

Android 以外の端末におけるウィンドウサイズの変更操作は Web ブラウザの resize イベントをフックすることによって取得している。resize イベントをフックした後に、window オブジェクトの innerWidth と innerHeight 要素の値を取得することによって、ウィンドウの横幅と高さを取得することができる。

一方、Android 端末においてはピンチ操作を行なってウィンドウのサイズ変更を行なっても Web ブラウザの resize イベントは発火しない。また、window オブジェクトの innerWidth と

innerHeight 要素は常にコンテンツの横幅と高さを示しているため、表示領域を特定するための要素としては利用することができない。そのため、画面の左上と右上、左下の3カ所に透明な div 要素を配置して、この div 要素の絶対座標を取得してから画面表示領域の幅と高さを計算している。

サーバにはウィンドウの横幅と高さの情報を送信している。サーバからリサイズイベントを受信したクライアント端末は、該当するフレームの大きさを変更する。

なお、フレームの頂点に対してドラッグ操作が行われた場合は、リサイズイベントをサーバに送信することによって実現している。

3.5 サーバサイド 通信部

サーバサイド通信部においては、現在接続している端末の情報を保持しており、ペアリング要求を受信した場合はこの情報を渡している。また、イベントを受信した際に、このイベントを現在接続している端末に向けて送信している。

第4章 利用シナリオ

本システムを利用する状況には様々な場合が考えられる。例えば、人数においては、一人のユーザによって利用する場合や、複数のユーザによって利用する場合が想定できる。また、ユーザが同じ場所に集まって作業をする場合や遠隔地に居るユーザ同士が作業をする場合が考えられる。

4.1 一人での利用例

4.1.1 小画面と大画面を連携して表示領域を拡張する利用例

Mさんはスマートフォンを用いて Web ブラウジングをしていた。Web ページがスマートフォンに対応していなかったため、横スクロールや拡大縮小操作を繰り返してコンテンツを閲覧していたが、閲覧にあまりに時間がかかり「こんなの、絶対おかしいよ。」と言って絶望した。そこで、Mさんは Interlocked Surfaces を用いてスマートフォンとノート PC のディスプレイを連携させた(図 4.1)。Mさんは広い表示領域を用いてコンテンツの閲覧をすることができたことに加えて、スマートフォン操作時に表示部を指で隠してしまうことなく閲覧をすることができた。

4.1.2 手元の画面と遠隔の画面を連携して遠隔操作を実現する利用例

Aさんは予め聴衆に配布した資料についてプレゼンテーションをしていた。配布資料について聴衆から質問があったため、スライドを操作する PC から離れて質問者の元へ行った。質問の回答に相応しいスライドを用意してあったため、表示をそのスライドに切り替えるために、PC まで戻らなければならなかった。そこで、発表者は Interlocked Surfaces を用いて発表者のスマートフォンとスライドを表示する PC を連携させておいた(図 4.2)。すると、発表者は PC から離れた状態でスライドを制御することができ、説明を円滑に行なうことができた。

4.2 複数人数での利用例

4.2.1 複数の画面を連携して共同作業を行う利用例

高校の古典部に所属する H さん、L さん、M さん、S さんの 4 名はテーブルを囲んで文化祭で出展する文集の作成のために情報収集を行っていた。4 名はそれぞれのノート PC を用



図 4.1: 小画面と大画面を連携して表示領域を拡張する利用例

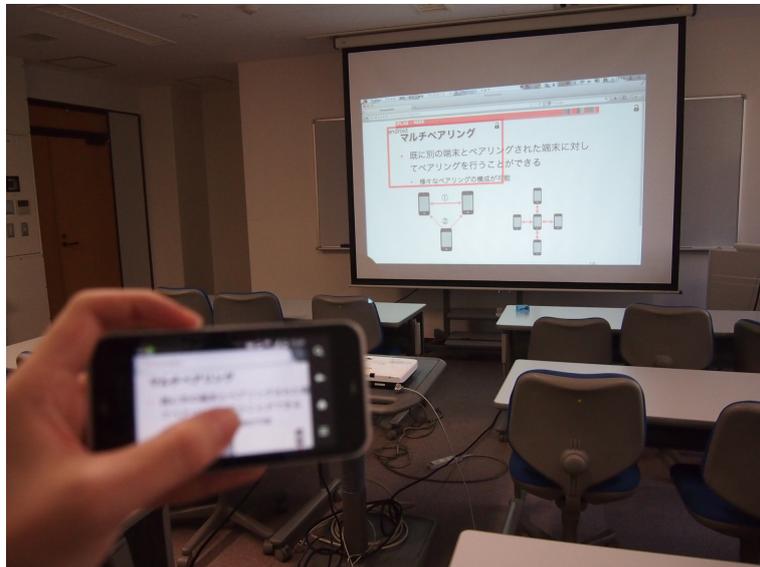


図 4.2: 手元の画面と遠隔の画面を連携して遠隔操作を実現する利用例

いて情報収集を行っていた。Oさんが有益なコンテンツを見つけたため、メンバーに共有しようとした。Oさんと隣り合っていたMさんとSさんは画面を一緒に見ることができたが、対面に座っていたLさんは画面を見ることができなかった。Lさんは「私、気になります。」と言って席を立った。

そこで、4人は Interlocked Surfaces を用いてそれぞれのノート PC をペアリングして作業をすることにした(図 4.3)。すると、有益なコンテンツをスムーズにメンバー間で共有することができ、さらに注目すべきポイントをカーソル共有を用いて指し示すことによって、情報共有を効果的に行なうことができた。



図 4.3: 複数の画面を連携して共同作業を行う利用例

4.2.2 遠隔地と画面を連携して共同作業を行う利用例

高校の軽音楽部に所属する Yさんと Mさんと Rさんは新曲の歌詞を考えるために音声チャットソフトとリモートデスクトップソフトウェアを利用して歌詞を書いていた。Mさんの PC がサーバマシンとなり、Yさんと Rさんは自分の PC から Mさんの PC にリモートアクセスしていた。3人はパートごとに分担をして歌詞の作成を行なおうとしたが、リモートデスクトップでは複数ユーザがテキストを同時入力することができなかった。そこで、3人は Interlocked Surfaces を用いてそれぞれの PC をペアリングして作業をすることにした。すると、分担して同時に歌詞を編集することに加え、他の人の編集内容をリアルタイムで見ることができたため、リアルタイムにフィードバックを行ない、スムーズに共同作業を進めることができた。

第5章 評価

提案システムを利用した感想を聴き、今後の改良と発展に繋げるために評価実験を行った。

5.1 実験目的

本実験の目的は情報の閲覧を行なう際にタブレットのみを利用した場合と、提案システムを利用した場合においてどのような違いが生じるか調査することである。

実験を行うにあたり、タブレット単体で情報を閲覧するよりも、他の大型のディスプレイと連携した方が情報の閲覧速度が向上するのではないかという仮説を立てた。

5.2 被験者

被験者は22～24歳の情報系の大学生と大学院生の計10名である。10名全員が日常的にスマートフォンを利用しており、タッチパネルにおける情報の閲覧操作は十分に慣れている。

5.3 実験内容

本実験では被験者がタブレットを操作し、2文字のアルファベットの組み合わせから指示された組み合わせを探し出すというタスクを行なってもらった。画面には2文字のアルファベットの組み合わせが昇目状に配置されている(図5.1)。アルファベットの組み合わせは規則的に並んでいる。左上を始点として右側に向かってAA, AB, AC, …, AZといった順番で並んでおり、下側に向かってAA, BA, CA, …, ZAといった順番に並んでいる。なお、マスは1辺が50ピクセルの正方形であり、コンテンツ全体の幅と高さは1300ピクセルである。

タスクを開始すると2文字のアルファベットを指示する指示画面が出現し、STARTボタンを押すとアルファベットの組み合わせを選択する選択画面へと遷移する(図5.2)。選択画面においては指示された組み合わせでないものを選択すると不正解であることを示すダイアログが出現し、指示された組み合わせを選択すると再び指示画面へと遷移する。このフローを1タスクにつき26回行なった。なお、アルファベットの組み合わせは、スクロール量に偏りが発生しないように考慮してある。

タスクにおいてはSTARTボタンを押してから指定されたターゲットを選択するまでの時間を記録した。また、実験終了後にアンケートを行なった。アンケートの項目は以下の通りである。

1. スマートフォンを用いてコンテンツを閲覧していると、画面サイズが小さくて見づらいつと感じることはあるか
2. 提案システムを利用した時の情報閲覧は快適であったか
3. 提案システムを利用しない時の情報閲覧は快適であったか
4. その他の意見や感想

なお、設問1～3は5段階のリッカート尺度と回答の理由を記述する欄を設けた。リッカート尺度は1が最も評価が低く、5が最も評価が高い。

実験には画面サイズが異なる2つのタブレットを利用した。1つはSHARP社のIS03¹であり、画面サイズは3.5インチである(図5.3左)。以降IS03を3.5インチタブレットと呼ぶ。もう1つはASUS社のNexus7²であり、画面サイズは7.0インチである(図5.3右)。以降Nexus7を7.0インチタブレットと呼ぶ。被験者10名を2つのグループに分け、5名は3.5インチタブレットを利用してタスクを行ない、残りの5名は7インチタブレットを用いてタスクを行った。

タスクにはタブレットのみを用いて上述したタスクを行なう条件Aと、タブレットと23インチディスプレイを提案システムを用いて連携させた状態でタスクを行なう条件Bがある。

5.3.1 条件A: タブレットのみ

被験者はタブレットのみを用いてタスクを行なう(図5.4)。本番タスクを行なう前に十分に操作に慣れるまで練習する時間を設けた。

5.3.2 条件B: タブレットと画面を連携

被験者はタブレットと23インチディスプレイを提案システムを用いて連携させた状態でタスクを行なう(図5.5)。なお、被験者には操作は全て手元のタブレットに対して行い、指示したアルファベットの組み合わせを探索する時に連携したディスプレイを利用するように指示した。本番タスクを行なう前に十分に操作に慣れるまで練習する時間を設けた。

5.4 実験結果

アンケートの結果をグラフ化したものを図5.6に示す。なお、エラーバーは標準偏差を示している。設問1は普段のスマートフォンの利用についての項目について問うものであるため、全被験者から得られたデータをまとめて扱っている。一方、設問2と設問3はタスクを行なっ

¹<http://www.sharp.co.jp/products/is03/>

²http://www.google.com/intl/ALL_jp/nexus/7/

| | | | | | | | | | | | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| AA | AB | AC | AD | AE | AF | AG | AH | AI | AJ | AK | AL | AM | AN | AO | AP | AQ | AR | AS | AT | AU | AV | AW | AX | AY | AZ |
| BA | BB | BC | BD | BE | BF | BG | BH | BI | BJ | BK | BL | BM | BN | BO | BP | BQ | BR | BS | BT | BU | BV | BW | BX | BY | BZ |
| CA | CB | CC | CD | CE | CF | CG | CH | CI | CJ | CK | CL | CM | CN | CO | CP | CQ | CR | CS | CT | CU | CV | CW | CX | CY | CZ |
| DA | DB | DC | DD | DE | DF | DG | DH | DI | DJ | DK | DL | DM | DN | DO | DP | DQ | DR | DS | DT | DU | DV | DW | DX | DY | DZ |
| EA | EB | EC | ED | EE | EF | EG | EH | EI | EJ | EK | EL | EM | EN | EO | EP | EQ | ER | ES | ET | EU | EV | EW | EX | EY | EZ |
| FA | FB | FC | FD | FE | FF | FG | FH | FI | FJ | FK | FL | FM | FN | FO | FP | FQ | FR | FS | FT | FU | FV | FW | FX | FY | FZ |
| GA | GB | GC | GD | GE | GF | GG | GH | GI | GJ | GK | GL | GM | GN | GO | GP | GQ | GR | GS | GT | GU | GV | GW | GX | GY | GZ |
| HA | HB | HC | HD | HE | HF | HG | HH | HI | HJ | HK | HL | HM | HN | HO | HP | HQ | HR | HS | HT | HU | HV | HW | HX | HY | HZ |
| IA | IB | IC | ID | IE | IF | IG | IH | II | IJ | IK | IL | IM | IN | IO | IP | IQ | IR | IS | IT | IU | IV | IW | IX | IY | IZ |
| JA | JB | JC | JD | JE | JF | JG | JH | JI | JJ | JK | JL | JM | JN | JO | JP | JQ | JR | JS | JT | JU | JV | JW | JX | JY | JZ |
| KA | KB | KC | KD | KE | KF | KG | KH | KI | KJ | KK | KL | KM | KN | KO | KP | KQ | KR | KS | KT | KU | KV | KW | KX | KY | KZ |
| LA | LB | LC | LD | LE | LF | LG | LH | LI | LJ | LK | LL | LM | LN | LO | LP | LQ | LR | LS | LT | LU | LV | LW | LX | LY | LZ |
| MA | MB | MC | MD | ME | MF | MG | MH | MI | MJ | MK | ML | MM | MN | MO | MP | MQ | MR | MS | MT | MU | MV | MW | MX | MY | MZ |
| NA | NB | NC | ND | NE | NF | NG | NH | NI | NJ | NK | NL | NM | NN | NO | NP | NQ | NR | NS | NT | NU | NV | NW | NX | NY | NZ |
| OA | OB | OC | OD | OE | OF | OG | OH | OI | OJ | OK | OL | OM | ON | OO | OP | OQ | OR | OS | OT | OU | OV | OW | OX | OY | OZ |
| PA | PB | PC | PD | PE | PF | PG | PH | PI | PJ | PK | PL | PM | PN | PO | PP | PQ | PR | PS | PT | PU | PV | PW | PX | PY | PZ |
| QA | QB | QC | QD | QE | QF | QG | QH | QI | QJ | QK | QL | QM | QN | QO | QP | QQ | QR | QS | QT | QU | QV | QW | QX | QY | QZ |
| RA | RB | RC | RD | RE | RF | RG | RH | RI | RJ | RK | RL | RM | RN | RO | RP | RQ | RR | RS | RT | RU | RV | RW | RX | RY | RZ |
| SA | SB | SC | SD | SE | SF | SG | SH | SI | SJ | SK | SL | SM | SN | SO | SP | SQ | SR | SS | ST | SU | SV | SW | SX | SY | SZ |
| TA | TB | TC | TD | TE | TF | TG | TH | TI | TJ | TK | TL | TM | TN | TO | TP | TQ | TR | TS | TT | TU | TV | TW | TX | TY | TZ |
| UA | UB | UC | UD | UE | UF | UG | UH | UI | UJ | UK | UL | UM | UN | UO | UP | UQ | UR | US | UT | UU | UV | UW | UX | UY | UZ |
| VA | VB | VC | VD | VE | VF | VG | VH | VI | VJ | VK | VL | VM | VN | VO | VP | VQ | VR | VS | VT | VU | VV | VW | VX | VY | VZ |
| WA | WB | WC | WD | WE | WF | WG | WH | WI | WJ | WK | WL | WM | WN | WO | WP | WQ | WR | WS | WT | WU | WV | WW | WX | WY | WZ |
| XA | XB | XC | XD | XE | XF | XG | XH | XI | XJ | XK | XL | XM | XN | XO | XP | XQ | XR | XS | XT | XU | XV | XW | XX | XY | XZ |
| YA | YB | YC | YD | YE | YF | YG | YH | YI | YJ | YK | YL | YM | YN | YO | YP | YQ | YR | YS | YT | YU | YV | YW | YX | YY | YZ |
| ZA | ZB | ZC | ZD | ZE | ZF | ZG | ZH | ZI | ZJ | ZK | ZL | ZM | ZN | ZO | ZP | ZQ | ZR | ZS | ZT | ZU | ZV | ZW | ZX | ZY | ZZ |

図 5.1: 升目状に配置されたアルファベットの組み合わせ

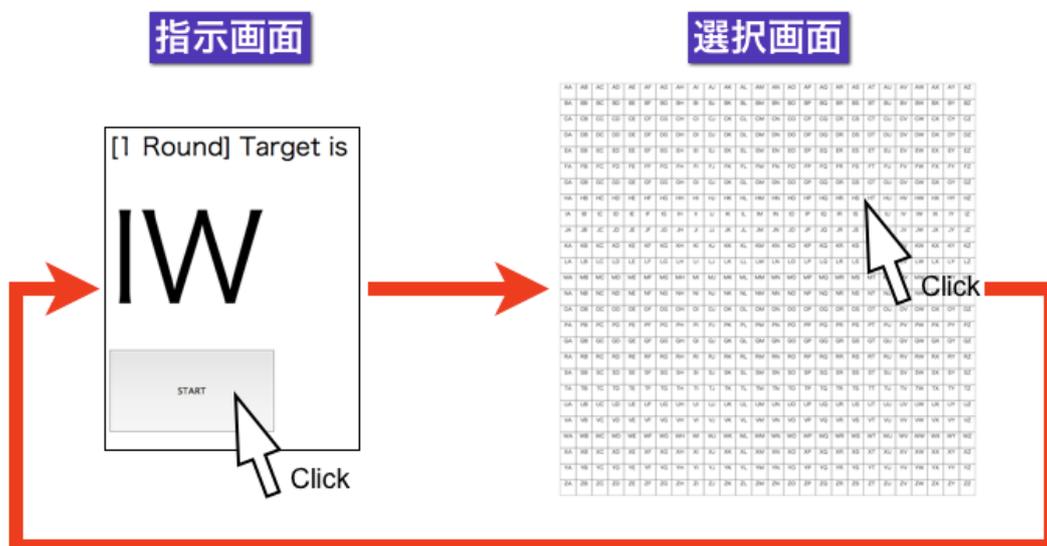


図 5.2: タスクの流れ



図 5.3: 実験に利用した端末



図 5.4: 条件 A においてタスクを遂行している様子

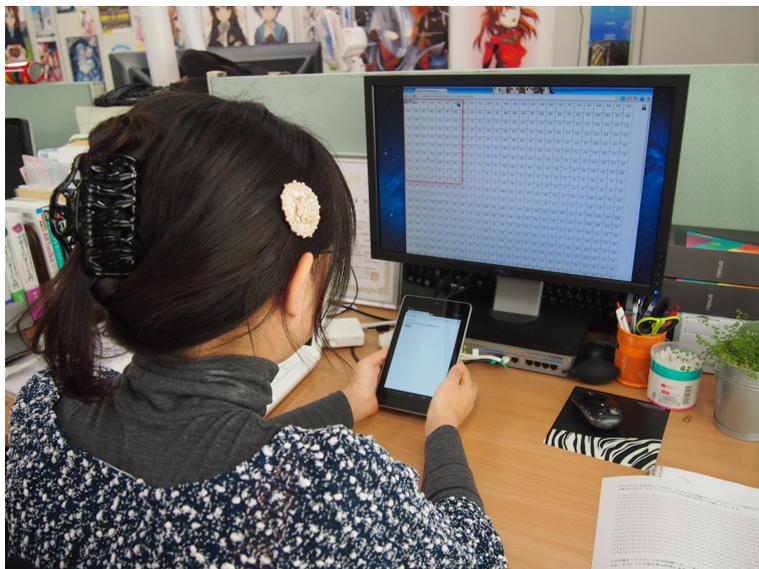


図 5.5: 条件 B においてタスクを遂行している様子

での利用感について問うものであるため、3.5インチ利用者と7.0インチ利用者のそれぞれにおいてグラフを分けて表示している。

設問1の『スマートフォンを用いてコンテンツを閲覧していると、画面サイズが小さくて見づらいつと感じることがあるか』という質問に対する回答の平均点は4.5であった。設問2の平均点は3.5インチを利用したグループと7.0インチを利用したグループ共に3.8であった。また、設問3の平均点は3.5インチを利用したグループが2.4であり、7.0インチを利用したグループは3.2であった。アンケートの定性的な評価の結果、システムを利用した場合の方がシステムを利用しない場合よりも操作を快適に行なえることがわかった。

次に、ターゲットを探し出すまでの時間についての結果を述べる。3.5インチタブレットを利用した場合のターゲットを探し出すまでにかかった時間の平均は図5.7の通りである。なお、縦軸はターゲットを探すまでにかかった時間をミリ秒によって示している。ターゲットまでの距離が600ピクセル未満の場合と、1200ピクセル以上の場合にはシステムの利用時と非利用時で有意差は無かったが、600ピクセルから1199ピクセルまでの場合、システム利用時の方が非利用時よりも有意に遅かった。

また、7.0インチタブレットを利用した場合のターゲットを探し出すまでにかかった時間の平均は図5.8の通りである。なお、縦軸はターゲットを探すまでにかかった時間をミリ秒によって示している。ターゲットまでの距離に依らず、全ての場合においてシステム利用時の方が非利用時よりも有意に遅かった。

最後に、それぞれの回答で得られたコメントの一部を示す。

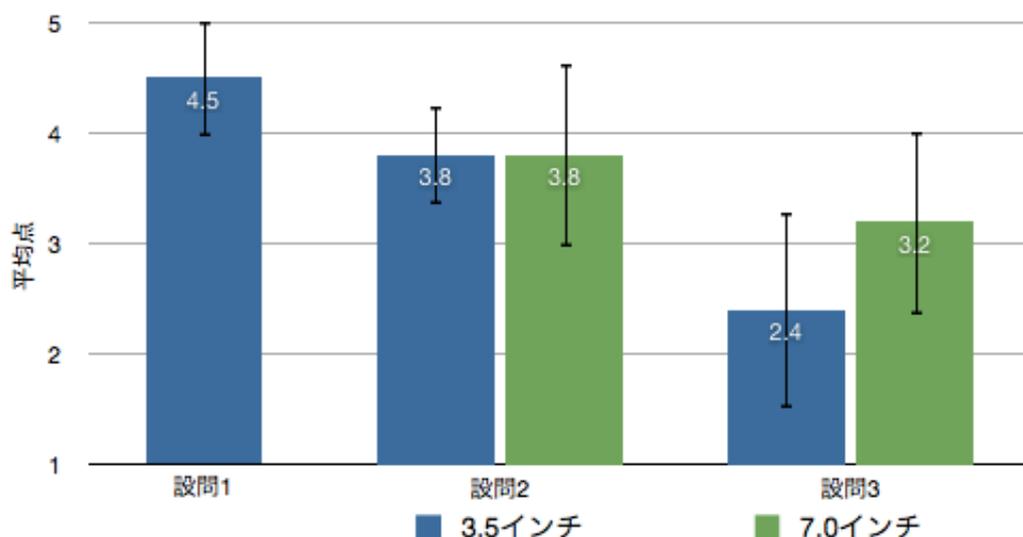


図 5.6: アンケートの結果

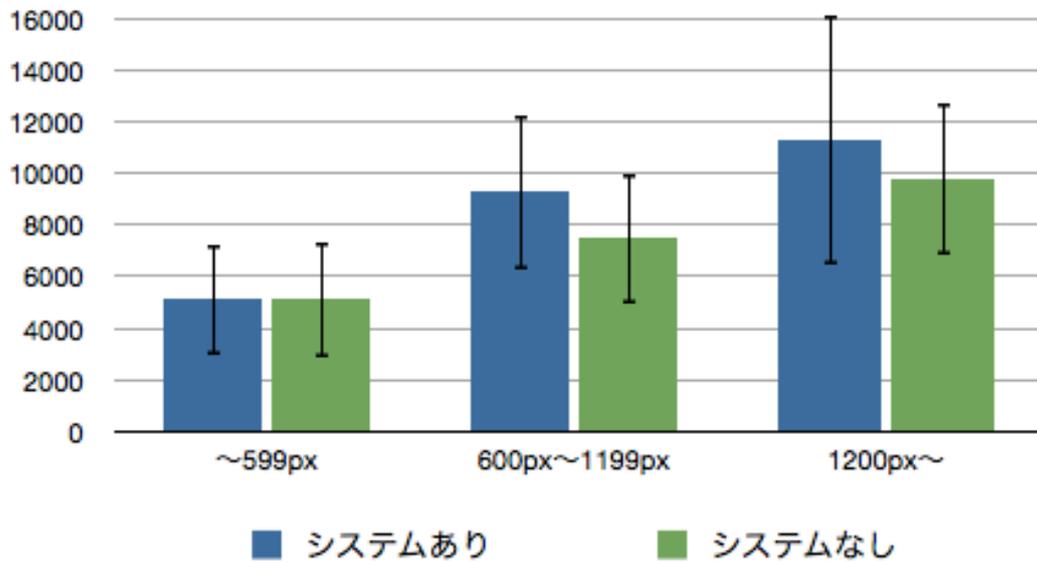


図 5.7: 3.5 インチタブレット利用時のターゲットを探し出すまでにかかった時間

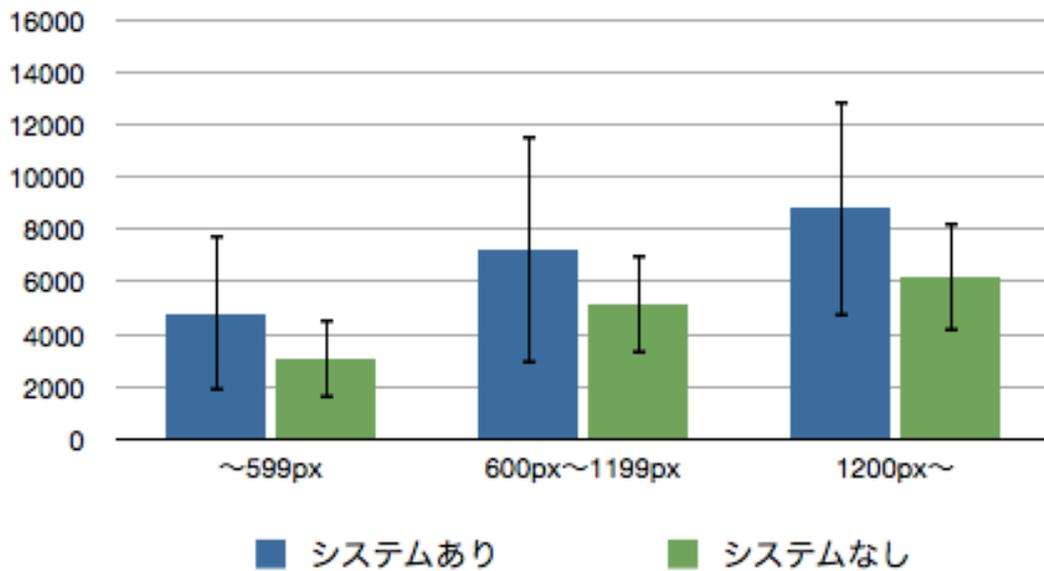


図 5.8: 7.0 インチタブレット利用時のターゲットを探し出すまでにかかった時間

設問2のコメント(一部抜粋)

7.0 インチタブレット利用者

- 全体が俯瞰できる画面の中で現在の画面位置がどの辺りなのかを把握できるため快適に操作できた。
- 手元のタブレットに対してスクロールを行なった際に、フレームの移動に反映されることが遅いことがあり気になった。
- 連携したディスプレイの画面が大きいため全体の把握が快適であった。
- 視線の移動と連携したディスプレイを見ながら操作を行なうことに少し不快感を感じた。
- ターゲットを探す際、表示が細かすぎて、自分が何処を見ているのかがわからなくなってしまうことがあった。
- スマートフォン側の操作が連携したディスプレイ側に反映されるまでにラグを感じるがあったので、操作が反映されているか確認しながらの利用になった。これが少し煩わしかった。

3.5 インチタブレット利用者

- 操作によって画面表示を隠してしまうことがなく快適に操作をすることができた。
- 閲覧が快適にできたが、操作のタイムラグによってスクロールしすぎてしまうことがあった。
- ターゲットを探すことを快適に行なうことができた。一方、視線移動があったため疲れを感じた。
- 連携した大きい画面で全体を把握することができたため、どこにあるのかが探しやすかった。
- ターゲットを探しやすかった。しかし、ディスプレイと端末の間で視線を動かす必要があるため効率が落ちることがあった。

設問3のコメント(一部抜粋)

7.0 インチタブレット利用者

- 普段利用しているスマートフォンよりも画面サイズが大きかったため、不快感をあまり感じなかった。

- 画面が小さかったと感じた。全体の把握のためにピンチズームを行なうことが煩わしいと思った。
- 普段からスマートフォンを利用しているため、ピンチ操作で大雑把に探してから細部を指定するという操作に慣れていてやりやすかった。
- 首や視点の移動が少ないため楽に感じた。

3.5 インチタブレット利用者

- 拡大しすぎるとターゲットが探しにくくなるので、あまり拡大せずに行なった。そうするとターゲットの大きさが小さくタッチが行ないにくかった。
- コンテンツがどこまで続くのか分かりにくいいため、スクロールしづらかった。
- いつも通りなので特に何も思わなかった。

設問4のコメント (一部抜粋)

7.0 インチタブレット利用者

- ターゲットのサイズが小さいものを選ぶ時にこのシステムを使いたいと思った。例えば、PC用のサイト(横幅900ピクセル以上あるもの)の中からテキストリンクを選択するような時に使いたい。
- もっと画面サイズが小さいスマートフォンで使ってみたいと思った。
- コンテンツ全体を閲覧する為にディスプレイを併用することはいいが、タブレットとディスプレイの間を視線移動する際に現在見ている箇所がどこかわからなくなり混乱することがあった。
- 配置が規則的であり場所を予測することができたため手間があまりかからなかった。ターゲットの場所が予測できないケースで使ってみたい。

3.5 インチタブレット利用者

- 目標が規則的に並んでいたため目標の場所が予測でき、あまり違いを感じなかったが、目標の場所がわからない際にはとても見やすく良いと思った。
- 利用時に視線移動が頻繁に発生するのは疲れを感じた。
- システムのレスポンスの速度が上がればより快適になりそうだと感じた。

- 手元のデバイスのスクロールが連携したディスプレイのフレームに反映されるまでに少し間があったため、システム利用時の方が時間がかかったと感じたが、使いやすさは利用時の方が良かった。
- 画面を閲覧するには問題無いが、選択の時に画面を見比べるのが大変だった。

5.5 考察

設問1の結果から多くのスマートフォンの利用者は画面サイズが小さいためコンテンツを見づらいと感じていることがわかった。そのため、表示領域を拡張して情報の閲覧効率を向上させるという本研究の1つの目的は十分に組み込む価値がある課題であると言える。

実験の結果、システムを利用することによって操作の快適性は向上するが、ターゲットを探し出すまでにかかる時間は長くなることがわかった。ターゲットを探し出すまでにかかる時間が長くなった原因は視線移動の問題とシステムにおける遅延の問題の2つが考えられる。

5.5.1 視線移動の問題

本システムを利用した場合、画面間の視線移動が2回発生する。1度目は最初にターゲットを確認してSTARTボタンを押し、そこからターゲットを探すために連携したディスプレイへ視線を動かす時であり、2度目はディスプレイ中のターゲットを発見し、そこからターゲットを選択するために手元のタブレットに視線を戻す時である。この視線移動の分だけ時間が長くかかったことが予想できる。また、フレーム表示によってタブレット中に表示されているターゲットの位置は大方予測可能であるが、視線移動後に再度ターゲットを探さなければならないため、時間が長くかかった可能性もある。

この問題を解決するために、複数画面を連携した場合にできるだけ端末間の視線移動を少なくする必要がある。連携したディスプレイを見ながら正確なポインティング操作を行なうためには、操作端末における指の位置をカーソルとしてディスプレイに表示させる方法が考えられる。

5.5.2 システムにおける遅延の問題

プロトタイプシステムの実装においてはサーバを利用したため、クライアントとサーバの間で情報を通信することに時間がかかる。また、ネットワークの遅延の影響を受けてしまう。そのため、操作端末に対してスクロールを行なった場合、ペアリング端末に表示されるフレームにスクロール操作が反映されるまでにタイムラグが生じることがある。このタイムラグによってターゲットを探す作業が中断してしまったり、ユーザが現在閲覧している位置を誤認識してしまったりすることによってターゲットを探すまでの時間が長くなってしまったと考えられる。

通信のタイムラグを短くするためには，サーバを利用せずに，P2Pによって情報の通信を行なう実装にするという改善策が考えられる。

第6章 関連研究

6.1 複数の端末を連携して1つの画面として活用する研究

複数の画面を連携して1つの画面として活用する研究は数多く行なわれている [4].

栗原ら [5][6] は複数の画面を連携させてプレゼンテーションを行なうプレゼンテーションツールを開発した. この研究においては複数の端末間でコンテンツを共有して表示している点において関連している. このシステムはプレゼンテーションというプレゼンターから聴衆に向けての一方向的な情報の伝達のために使われるため, プレゼンターが全画面の操作権を持っている. しかし, 本研究は複数人数における共同作業においても利用できることを想定しており, 全ての端末のユーザが自身の端末に対して操作を行なうことができる.

Nam ら [7] はタイル状に設置した高解像度ディスプレイ同士を連携させて1つの大画面と同様に扱うシステムを開発した. 複数の画面においてコンテンツを共有して表示させる点において本研究として異なるが, この研究は情報の提示に焦点を当てている点が異なる.

6.2 複数の端末を連携してそれぞれ独立して活用する研究

把持したスマートフォンの画面中のカメラ映像に対して操作を行うことによって, 別の情報端末を遠隔操作することができるシステム Touch Projector [8] や, マイクロプロジェクタのメタファを用いてスマートフォンとディスプレイ間の連携を行ない, 情報の閲覧や操作を行なうことができるシステム Virtual Projection[9] などがある. これらの特徴は小型の個人端末と中・大型の公共端末を連携することによって, 表示領域の拡張や, 複数人数における情報閲覧を円滑化できることである. 本研究においても個人端末と公共端末を連携することの利点を活用する. さらに, 個人端末同士の連携もハイブリットに行なうことによって複数人数における情報閲覧と操作を円滑化できるシステムを構築する.

Boring ら [10] は携帯端末のカメラ映像越しに別の端末をタップすることによってその情報を携帯端末に転送することができるシステム Shoot & copy を開発した. このシステムはパブリックスペースの大画面に映し出された情報を個人所有の端末が取得できるようにすることを目的としており, 個人所有の端末の情報を大画面に映し出すことはしていない. それに対して本研究は個人所有端末と大画面の間で相互に情報のやり取りを行う事ができる.

Chang ら [11] は携帯端末のカメラでデスクトップの画面を撮影するという動作で2つの端末間で情報を通信するシステム Deep Shot を開発した. Deep Shot は携帯端末とデスクトップ PC 間の作業の移行を容易にする. 例えば, デスクトップ PC で表示されている情報を携帯端

末で閲覧したいとき、携帯端末のカメラでデスクトップを撮影することによってその情報を移行することができる。また、携帯端末からデスクトップへの情報の移行も同様に行う事ができる。Deep Shot では携帯端末とデスクトップの間での作業の移行をスムーズに行うことを目的としていることに対し、本研究では2つの端末を同時に使う事によって情報の閲覧を支援する事を目的としている。

角ら [12] は複数端末間でコンテンツを共有し、それぞれの端末からコンテンツに対して書き込みを行なえるシステム PhotoChat を開発した。この研究においては各ユーザが行なうことができるインタラクションはコンテンツに対しての書き込みに限られている。それに対して本研究は書き込みだけでなくスクロール等様々なインタラクションを行なうことができる。

6.3 複数の端末を連携するインタラクションについての研究

Schmidt ら [13] はスマートフォンの角をテーブルトップの面に当てることによって2つの端末を連携するインタラクション手法を提案した。このインタラクションは連携対象を直感的に指定できる点において優れている。しかし、本研究のように遠隔に存在する端末同士を連携させるには適していない。

Seewoonauth ら [14] は携帯端末と PC を接触させることによってデータの選択やデータの転送ができるシステム Touch & Connect and Touch & Select を開発した。この研究は複数端末間でのデータのやり取りに着目をしていることに対して、本研究はデータのやり取りだけでなく操作を行なうことができる。

第7章 議論

複数端末連携インタフェースを実現した Interlocked Surfaces は現在の実装においても一人のユーザ、または複数人数のグループにおいて十分に実用的であるが、プロトタイプシステムであるが故に何点か改善が必要である点がある。最も改善が必要なのは、端末間の情報転送にタイムラグが発生してしまう点である。プロトタイプシステムにおいてはサーバを用意したため、クライアントとサーバの情報通信に時間がかかってしまう。現在の実装においても通信する情報の量を最小限にする工夫をしているが、更にタイムラグを軽減するためには、サーバを利用せずに端末間で P2P による通信を行なうように実装を改良する必要がある。しかし、近年ネットワーク技術が進歩してきており情報の転送速度は徐々に早くなってきているため、将来的にはこの問題は解決される可能性がある。

また、評価実験を行なったことによって端末を連携した際に、端末間の視線移動はユーザの疲労感とポインティングの速度が落ちることがわかった。そのため、できるだけ端末間において視線移動を行なわなくても良いようにインタフェースの改良を検討する必要がある。

最後に、カメラを用いたペアリング手法にも課題がある。現在は AR マーカを表示して認識することによって対象端末の識別と位置姿勢の検出を行なっているが、AR マーカを表示することによって表示中のコンテンツを隠してしまうという問題や、光源の問題によって AR マーカを認識しづらいという問題がある。最近ではイメージベースの AR ライブラリが登場している。表示されているコンテンツ自体を画像マーカとして利用することによってこの問題を解決することができる可能性がある。

第8章 結論

本研究では複数情報端末を連携する際の(1)連携できる端末数の制限、(2)接続切り替えについての問題、(3)セキュリティの問題という3つの問題点を解決し、かつ様々な複数の情報端末同士を動的に連携することによって、情報の閲覧と操作を円滑化するための複数画面連携インタフェースを提案し、プロトタイプシステムとして *Interlocked Surfaces* を開発した。

まず、連携できる端末数の制限については、ケーブルは用いずに無線ネットワークを利用しているため、サーバの負荷が限界に達しない限り何台でも接続することが可能である。

次に、接続切り替えについての問題については、カメラ越しに連携する対象を選択するというインタラクション手法によって直感的かつケーブルを差し替えるといった手間無く接続切り替えを実現することが可能である。

最後に、セキュリティの問題については、端末に保存された情報は一切共有せず、コンテンツと画面位置の情報のみを共有することによって、他人が端末にアクセスすることによる情報漏洩などの危険性を排除した。

また、プロトタイプシステムの有用性を検証するために評価実験を行なった。実験においては1人のユーザが小画面と大画面を連携することによって操作が快適になるか、情報探索の時間が短縮されるかということを検証した。その結果、システムを利用した場合、情報探索の快適さは向上したものの、探索する時間は長くなった。探索時間が長くなってしまった原因は、視線移動が発生する問題と、タイムラグの問題であると考えられる。今後は視線移動を最小限とするインタフェースの改良を検討し、タイムラグ対策としてはP2P通信を用いた実装に改善する予定である。

謝辞

本論文の執筆にあたり，ご指導頂きました田中二郎教授に心より感謝申し上げます。

参考文献

- [1] Yuji Ayatsuka, Nobuyuki Matsushita, and Jun Rekimoto. New Real-World Oriented User Interfaces with “Gaze-Link” Metaphor. *Interaction 2000*, pp. 181–188, 2000.
- [2] 綾塚祐二, 松下伸行, 暦本純一. Gaze-Link:実世界指向ユーザインタフェースにおける「見ているものに接続する」というメタファ 特集 次世代インタラクションのための情報技術 . 情報処理学会論文誌, Vol. 42, No. 6, pp. 1330–1337, 2001.
- [3] 加藤博一. 拡張現実感システム構築ツール ARToolKit の開発. 電子情報通信学会技術研究報告. PRMU, パターン認識・メディア理解, Vol. 101, No. 652, pp. 79–86, 2002.
- [4] Tao Ni, G S Schmidt, O G Staadt, M A Livingston, R Ball, and R May. A Survey of Large High-Resolution Display Technologies, Techniques, and Applications. In *Proc. Virtual Reality Conference, 2006*, pp. 223–236, 2006.
- [5] Kurihara Kazutaka and Igarashi Takeo. A Flexible Presentation Tool for Diverse Multi-display Environments. *Proc. INTERACT 2007*, pp. 430–433, 2007.
- [6] Kazutaka Kurihara, Toshio Mochizuki, Hiroki Oura, Mio Tsubakimoto, Toshihisa Nishimori, Jun Nakahara, Yuhei Yamauchi, and Katashi Nagao. Linearity and synchrony: quantitative metrics for slide-based presentation methodology. In *Proc. ICMI-MLMI 2010*, pp. 33:1—33:4. ACM, 2010.
- [7] Sungwon Nam, Sachin Deshpande, Venkatram Vishwanath, Byungil Jeong, Luc Renambot, and Jason Leigh. Multi-application inter-tile synchronization on ultra-high-resolution display walls. In *Proc. MMSys 2010*, pp. 145–156. ACM, 2010.
- [8] Sebastian Boring, Dominikus Baur, Andreas Butz, Sean Gustafson, and Patrick Baudisch. Touch projector: mobile interaction through video. In *Proc. CHI 2010*, pp. 2287–2296. ACM, 2010.
- [9] Dominikus Baur, Sebastian Boring, and Steven Feiner. Virtual projection: exploring optical projection as a metaphor for multi-device interaction. In *Proc. CHI 2012*, pp. 1693–1702. ACM, 2012.

- [10] Sebastian Boring, Manuela Altendorfer, Gregor Broll, Otmar Hilliges, and Andreas Butz. Shoot & copy: phonecam-based information transfer from public displays onto mobile phones. In *Mobility 2007*, pp. 24–31. ACM, 2007.
- [11] Tsung-Hsiang Chang and Yang Li. Deep shot: a framework for migrating tasks across devices using mobile phone cameras. In *Proc. CHI2011*, pp. 2163–2172. ACM, 2011.
- [12] Yasuyuki Sumi, Jun Ito, and Toyoaki Nishida. Photochat: communication support system based on sharing photos and notes. In *Proc. CHI '08 Extended Abstracts on Human Factors in Computing Systems, CHI EA '08*, pp. 3237–3242. ACM, 2008.
- [13] Dominik Schmidt, Julian Seifert, Enrico Rukzio, and Hans Gellersen. A cross-device interaction style for mobiles and surfaces. In *Proc. the Designing Interactive Systems Conference, DIS '12*, pp. 318–327. ACM, 2012.
- [14] Khoovirajsingh Seewoonauth, Enrico Rukzio, Robert Hardy, and Paul Holleis. Touch & connect and touch & select: interacting with a computer by touching it with a mobile phone. In *Proc. MobileHCI 2009*, pp. 36:1–36:9. ACM, 2009.