

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

携帯デバイスを入力装置として用いた公共空間
の大画面インタラクション手法

金春明

(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、電車の駅やショッピングモールのような公共空間での壁サイズ大画面の利用が増えている。しかし、現在の壁サイズ大画面はユーザからの操作ができず、一方的にユーザに情報を提示するだけである。一方、携帯電話や PDA のような携帯デバイスの利用が急速に増えている。このような携帯デバイスは持ち歩くのが便利であるが、画面が小さくて一画面に表示できる情報の量が限られている。

我々はこのような問題に対し、ユーザが公共空間の壁サイズ大画面と情報交換することを可能にし、大画面の広い画面で携帯デバイスの小画面問題を解決することを試みる。本研究では、携帯デバイスを公共空間の壁サイズの入力装置として用いて、壁サイズ大画面での操作を行うようなインタラクション手法を提案し、プロトタイプの実装を行った。さらに、QR コードを用いて壁サイズ大画面と携帯デバイスの接続を行った。ユーザは携帯デバイスのカメラで QR コードを撮ることだけで、壁サイズ大画面との接続をすることができる。それから、壁サイズ大画面に大量の情報が提示されると想定し、それらの情報から、ユーザにとって有用な情報を早く見つけることを支援する MARU インタフェースの開発を行った。

目次

第 1 章	はじめに	1
1.1	研究の背景.....	1
1.2	研究の目的.....	1
1.3	論文の構成.....	2
第 2 章	関連研究	3
2.1	大画面とのインタラクション手法に関する研究.....	3
2.2	大画面と携帯デバイスの接続に関する研究.....	4
2.3	大画面での操作に関する研究.....	4
第 3 章	壁サイズ大画面と携帯デバイスのインタラクション	7
3.1	QR コードを用いた大画面と携帯デバイスの接続方法.....	8
3.2	個人領域	9
3.3	操作インタフェース	12
3.3.1	携帯デバイス側の操作インタフェース	12
3.3.2	大画面側の操作インタフェース：MARU インタフェース.....	15
第 4 章	IWS の実装	21
4.1	システムの構成.....	21
4.2	サーバ側の実装.....	22
4.2.1	個人領域の実装.....	23
4.2.2	操作の実行の実装.....	24
4.2.3	MARU インタフェースの実装.....	27
4.3	ユーザ側の実装.....	29
4.4	iWS の使用例	30
4.4.1	電車の駅での利用例	30
4.4.2	ショッピングモールでの利用例	31
第 5 章	IWS の考察	32
5.1	提案システム iWS の利点	32
5.2	今後の改善点	32
5.2.1	プライバシーの問題	32
5.2.2	セキュリティ問題.....	33
5.2.3	大画面に公開された情報の管理	33
第 6 章	おわりに	34

謝辭.....	35
参考文献.....	36

図目次

図 2-1	Vacuum インタフェース	5
図 2-2	Frisbee インタフェース	6
図 3-1	システム概要	7
図 3-2	大画面での QR コードの配置	9
図 3-3	大画面に表示された個人領域	10
図 3-4	個人領域に表示された携帯デバイス内の情報	11
図 3-5	大画面と個人領域間の情報やり取り	12
図 3-6	違うイメージを持った仮想カーソル	13
図 3-7	PS モードインタフェースの動き方	13
図 3-8	CS モードインタフェースの動き方	14
図 3-9	ジャーのふたとペンの動きの関係	15
図 3-10	MARU インタフェース	16
図 3-11	MARU インタフェースで情報収集する操作	17
図 3-12	MARU インタフェースで収集した情報を Explorer に表示する	18
図 3-13	MARU インタフェースの情報収集方法 (a)	19
図 3-14	MARU インタフェースの情報収集方法 (b)	19
図 3-15	MARU インタフェースの情報収集方法 (c)	19
図 3-16	二つの MARU インタフェースが隣接した場合	20
図 4-1	システム構成図	21
図 4-2	大画面と携帯デバイスの接続を行う様子	23
図 4-3	大画面で個人領域を出した様子	24
図 4-4	PS モードインタフェース	25
図 4-5	CS モードインタフェース	26
図 4-6	ドラックイベントが発生時、仮想カーソルのイメージを変える	27
図 4-7	マルドーナツとマルディスクの半径	28
図 4-8	MARU インタフェースの移動	28
図 4-9	MARU インタフェースを利用している様子	29

第1章 はじめに

1.1 研究の背景

近年、公共の場での壁サイズ大画面の利用が増えてきている。壁サイズの大画面は画面が非常に大きくて、動的に情報をユーザに提供することができるので、同時に大勢の人に情報を提供するの非常に有効である。そのため、電車の駅で電車の時刻案内や駅周辺の道案内などに使われたり、ショッピングモールで商品案内に使われたりする。しかし、現在の大画面はただ、一方的にユーザに情報を提供していて、ユーザ側からの操作は非常に困難である。それにより、あるユーザに対してそのユーザにとって有用な情報だけを提示するのが困難である。

一方、携帯電話や PDA(Personal Digital Assistant)などの携帯デバイスは非常に小さくて持ち歩きやすいので、ユーザは場所と時間の制限を受けず、どこでも、いつでも情報を処理することができる。さらに、このような携帯デバイスの多機能化により、携帯デバイスを用いて写真の撮影をしたり、インターネットをアクセスしたりすることもできるようになった。しかしながら、このような携帯デバイスは画面も小さいので、一画面に表示できる情報の量が非常に限られている。よって、携帯デバイスを利用して多くの情報を閲覧するのが困難である。携帯デバイスの小画面問題を解決するためにいろんな研究[3]がなされているが、携帯デバイス自体では解決できない。なぜなら、携帯デバイスは小さいのがメリットであるが、画面を大きくするとデバイスもそれに従って大きくなるので、携帯デバイスの小さいというメリットがなくなる。

1.2 研究の目的

本研究では壁サイズの大画面と携帯デバイスをうまく組み合わせ、ユーザが壁サイズの大画面での操作を可能にし、携帯デバイスの小画面問題を解決しようとする。これにより、ユーザが携帯デバイスを用いて、大画面で操作を行い、さらに大画面と携帯デバイス間の情報交換ができるようにする。最終的には大画面がただの情報提示ではなく、ユーザたちの情報交換の場となり、ユーザが自分の持っていた情報を大画面に公開することや大画面から自分にとって有用な情報を取得することが簡単にでき、さらに大画面の広い画面を携帯デバイスの小さい画面の代わりに使えるようにすることが本研究の目的である。

壁サイズ大画面と携帯デバイスを組み合わせて利用しようとする以下の問題を考える必要がある。

- 1 大画面と携帯デバイスの接続方法
- 2 大画面との情報交換方法
- 3 大画面での操作

大画面と携帯デバイスの接続方法はいろいろとある。研究[10][12]のように物理的に接続する方法が考えられる。しかし、物理的に接続するといろんな制限が与えられ、ユーザが大画面での利用の上の自由度が下がる。例えば、ユーザは大画面から大画面と接続するためのケーブルの長さ以上は離れない。複数のユーザが使っている時、ケーブルの管理が困難であると考えられる。そして、物理的に接続すると同時に使えるユーザの数が決められてしまい、ユーザ数を動的に管理することが困難である。それから、このような接続方法は操作が煩わしい。そのため、大画面と携帯デバイスの接続方法としてはワイヤレスネットワークを通じて接続する方法が一番よいであり、さらに、ワイヤレスネットワークを通じた接続が簡単にできるようにする必要が考えられる。

現在、公共空間に設置してある大画面とユーザのデバイスの間での情報交換はできなくなっている。もし、大画面とユーザのデバイス間での情報交換ができるのであれば、大画面をもっと有効に利用することができる。例えば、大画面に表示した情報をユーザが自分のデバイスに保存すると、大画面から離れてもその情報を見ることができる。さらに、このように情報をユーザのデバイスに保存すると、その情報に基づいてもっと多くの情報に辿って行くことができるようになる。そして、ユーザからも大画面に情報を公開することができて、ユーザが他のユーザとコミュニケーションをもっと有効に支援することができる。それから、ユーザが直感的で簡単に大画面との情報交換を行えるインタフェースが必要である。

現在の公共空間の大画面はほとんどユーザからの操作ができなくなっている。わずかにユーザからの操作ができる大画面は、タッチパネルを利用するのが多い。しかし、タッチパネルつきの大画面は高価であるや、画面が大きくなるとユーザの手が届かないところは操作できないなどの欠点がある。なので、大画面での操作は直接画面に操作する方法ではなく、ユーザが気軽に持ち歩ける携帯デバイスを入力装置として利用して、大画面での操作を行う方法が良いと考えられる。さらに、大画面の大量の情報からユーザにとって有用な情報をすばやく見つけ出せることを支援する必要があると考えられる。

これらの問題に対し、本研究では以下のような方法で解決を試みる。壁サイズ大画面と携帯デバイスの接続は QR コードを用いて接続を行う。ユーザが携帯デバイスで QR コードを撮る操作をするだけで接続ができるようにする。大画面と携帯デバイスの情報交換手法としては大画面の一部分をユーザの個人領域としてユーザに提供し、ユーザは個人領域で携帯デバイスにある情報を表示したり、大画面で提供した情報を閲覧したりすることができるようにする。さらに、大画面と携帯デバイスの情報交換はドラッグ操作で簡単にできるようにする。最後に大画面での操作に関しては、ユーザの携帯デバイスを大画面の入力装置として用いて大画面での操作を行う。大画面にある大量情報の閲覧を支援するためには、ユーザから近い位置に表示された情報を集めて表示するような MARU インタフェースを提案する。

1.3 論文の構成

本論文の構成は以下のとおりになっている。

第 2 章で大画面のインタラクション、大画面と携帯デバイスの接続、大画面での操作の三つのカテゴリに分けて、それぞれの関連研究について述べる。第 3 章では提案システムの構成や使い方について述べる。第 4 章では提案システム iWS の実装について述べる。第 5 章では提案システムに対する議論を述べ、第 6 章でまとめる。

第2章 関連研究

本章では、研究の目的で述べた 3 点の考慮すべき問題に関する関連研究を述べる。

2.1 大画面とのインタラクション手法に関する研究

Jun Rekimoto の研究 [1] では、Pick-and-Drop という手法を提案している。Pick-and-Drop は PDA やタブレット PC などの携帯デバイスの画面からスタイラスを利用してオブジェクトを取って、他のコンピュータの画面にドロップすることで、携帯デバイスのオブジェクトを他のコンピュータに転送する技術である。これは従来の Drag-and-Drop の拡張バージョンでドラッグしたオブジェクトを他のコンピュータにドロップすることができる。この手法は操作が非常に直感的で、ユーザにとって理解や操作しやすい。しかし、この手法はコンファレンスルームのような比較的にユーザが少ない場所での利用に適している。そして、この手法の欠点は利用できる画面のサイズが人の手の届ける範囲内である制約を持っていることである。

Dynamo[10][19]では、ユーザが大画面の一部分を個人領域として使えることができ、さらにユーザ自身の USB ディスクなどのメモリ装置を大画面に繋げて、大画面と情報交換することができる。さらにマルチユーザシステムになっているので、複数のユーザが同時に大画面を利用することができる。論文では PDA などの携帯デバイスと連携して使えると言っていたが、デバイスの接続方法を記述せず、基本的にマウスとキーボードを直接大画面に接続し、それらによって大画面での操作を行う。このような接続方法や操作方法はコンファレンスルームのような場所に適していて、駅やショッピングモールなどの公共の場では利用できない。

他に iRoom[11]のような研究がある。これらの研究は大画面を携帯デバイスと連携して利用しようとするが、その想定している利用空間がコンファレンスルームのような限られたユーザが利用できる場所である。そのため、これらのインタラクション手法は駅などの誰でも入れる、つまり、大画面を誰でも利用できるような環境で利用するのは困難である。

駅などの公共空間でのインタラクションはまず、コンファレンスルームのような限られたユーザだけが使う環境ではなく、誰でも、且つ、大勢のユーザが同時に使うような環境を考慮する必要がある。そして、大画面のサイズが非常に大きいので、手が届かない所があるということも考慮すべきである。

2.2 大画面と携帯デバイスの接続に関する研究

大画面と携帯デバイスを連携して利用するためには、まず大画面と携帯デバイスを接続する必要がある。WebWall[6]では携帯電話を利用し、喫茶店などに設置された大画面とインタラクションをする手法を提案している。ユーザは携帯電話で大画面に情報を送り、大画面でその情報を映し、その場にいる他の人たちと楽しむ事ができる。情報を携帯電話から大画面に送るには、携帯電話の通信網を利用する。携帯電話で何らかのメッセージを作成し、大画面に表示してある番号に送る。そうすると、メッセージは携帯電話の通信網を通じ、その番号に対応するメッセージキューに入る。大画面に空きがある時にメッセージキューから順番にメッセージを取り出し、大画面に表示する。この手法だと、ユーザへのフィードバックが遅い。大画面に自分のメッセージが表示されるまで、大分待たなければならない。それから、ユーザが送信してから画面に表示するまで時間が掛かるので、リアルタイムで情報を表示することができない。あるユーザが大画面を見て、自分の考えを大画面に送信したのに、表示された時は、話題がすでに他の話題になっていたりすることがありうる。

それから、Brad[12]らは物理的に複数の PDA を一つの PC に接続する試みをしたが、公共空間では有効な方法ではない。その他に Bluetooth を利用して接続する方法[4],[5]も提案されている。

公共空間で大画面と携帯デバイスを接続するにはワイヤレスネットワークを通じて接続し、ユーザにとって分かりやすい、操作しやすい、且つ、フィードバックの早い方法が望ましい。

2.3 大画面での操作に関する研究

大画面での操作に関しても、色んな研究がなされている。Rafael Ballagas[2]らはカメラ付の携帯電話を用いて、大画面でのポインティングを行い、操作をする。操作の仕方はカメラを開き、大画面の画面を撮れるようにし、カーソルを移動させたい方向に移動する。そうすると、カーソルが同じ方向に移動する。イベントは携帯電話のボタンを押すことで発生させる。携帯電話のカメラで大画面の画面を撮った画像とその次に撮った画像を比較し、それにより携帯電話の移動方向や移動量を計算する。その値を用いてカーソルをコントロールする。この方法は非常に直感的である。しかし、携帯電話を操作が終わるまで、大画面に向けて移動させる必要があるため、ユーザの腕に負担が掛かってしまう。そのため、長時間の操作はできない。

C-Blink[13]では携帯電話の画面を利用し、携帯電話の画面を大画面に向けて点滅させることで大画面でのポインティングを行う。大画面の上にはイメージセンサーとしてカメラを設置し、携帯電話の画面をカメラに向けて色を変えながら点滅させることで 8 ビットの信号を送る。信号としてユーザの ID やイベントの ID などを送ることができるので、大画面で特定のユーザが操作を行うことが可能になる。しかしながら、この手法もユーザの腕に負担が掛かってしまう欠点がある。

それから、専用のデバイスを利用して大画面での操作を行うような手法[14]も提案されている。しかし、公共空間でユーザに大画面操作のための専用デバイスを使わせるのは望まし

くない。ユーザがすでに持っているデバイス、つまり、携帯電話や PDA のような携帯デバイスを利用して大画面での操作を行うのが望ましい。それから、できるだけユーザに負担がかからない方法が良いであろう。

他に大画面での操作手法に関する研究として、Vacuum[16]、Frisbee[17]などがある。Vacuum (図 2-1) では大画面の画面の広さを考慮して、arc edge で囲まれた領域内にあるオブジェクトを bull's-eye の近くに表示させる。それから、Frisbee(図 2-2)では Local Space と Remote Space があって、Local Space のなかに Remote Space の画面を表示する。それから、二つのコンポーネントはお互いに遠く配置される。Remote Space のオブジェクト (図 2-2 の a) を Local Space からドラックして出す (図 2-2 の b) ことで、Remote Space のところにあるオブジェクトを Local Space のところにドラックしてきてることができる。逆に、Local Space のところにあるオブジェクトを Remote Space のところにドラックしていくこともできる (図 2-2 の c と d)。それから、研究[9]のように、大画面の大きさを考慮し、遠いところにあるオブジェクトを近くに持ってきて操作すると良い。

しかし、Vacuum インタフェースは arc edge の範囲が大きすぎるので、大量の情報がある場合には逆に使いにくくなる傾向があると考えられる。そのために、その範囲をユーザの周りの一定の領域に指定した方が良いであろう。Frisbee インタフェースは個々のオブジェクトに対して操作を行い、複数のオブジェクトを一遍に見るには不便であると考えられる。(図 2-1 は論文[16]の 363 ページの図を引用した。図 2-2 は論文[17]の 129 ページの図を引用した。)

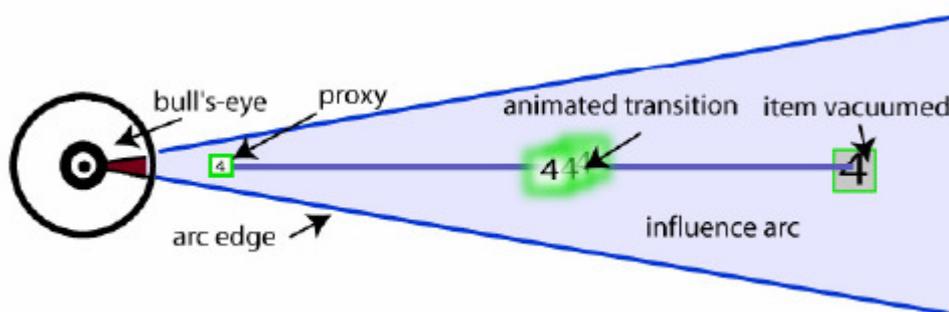


図 2-1 Vacuum インタフェース

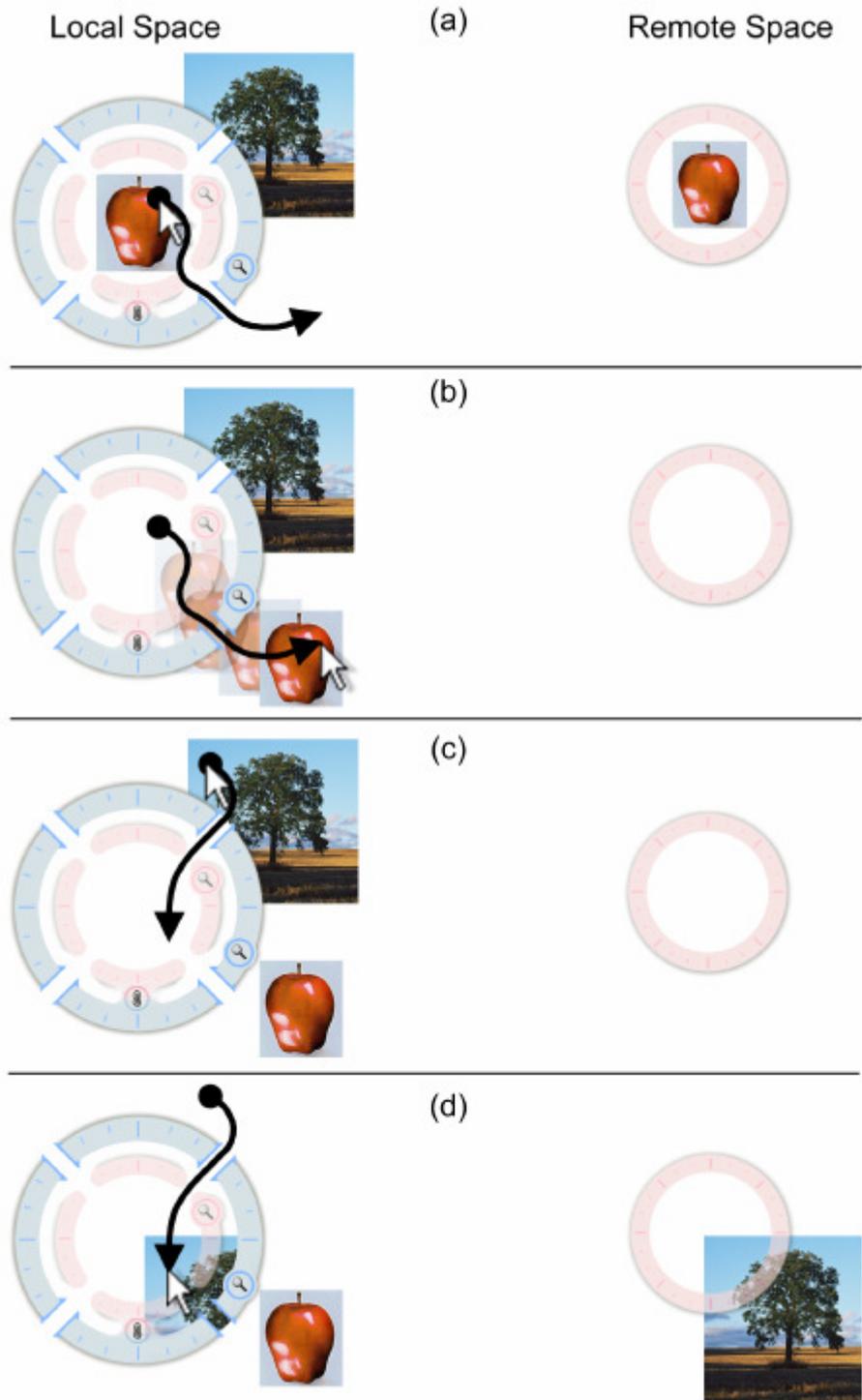


図 2-2 Frisbee インタフェース

第3章 壁サイズ大画面と携帯デバイスのインタラクション

本研究では携帯デバイスを公共空間の大画面の入力装置として用いて、大画面とのインタラクションを行う手法を提案する[8]。さらに、大画面はユーザに個人領域を提供し、ユーザが個人領域を通じて、大画面とのインタラクションをすることができる。このようなインタラクション手法により、ユーザは携帯デバイスでの操作で大画面での操作を行うことができ、大画面に携帯デバイスにある情報を出して、携帯デバイスの小さい画面ではなく、大画面の広い画面で情報に対して操作をすることができるようになる。さらに、個人領域を利用することで、直感的に大画面との情報交換を行うことができるようになる。

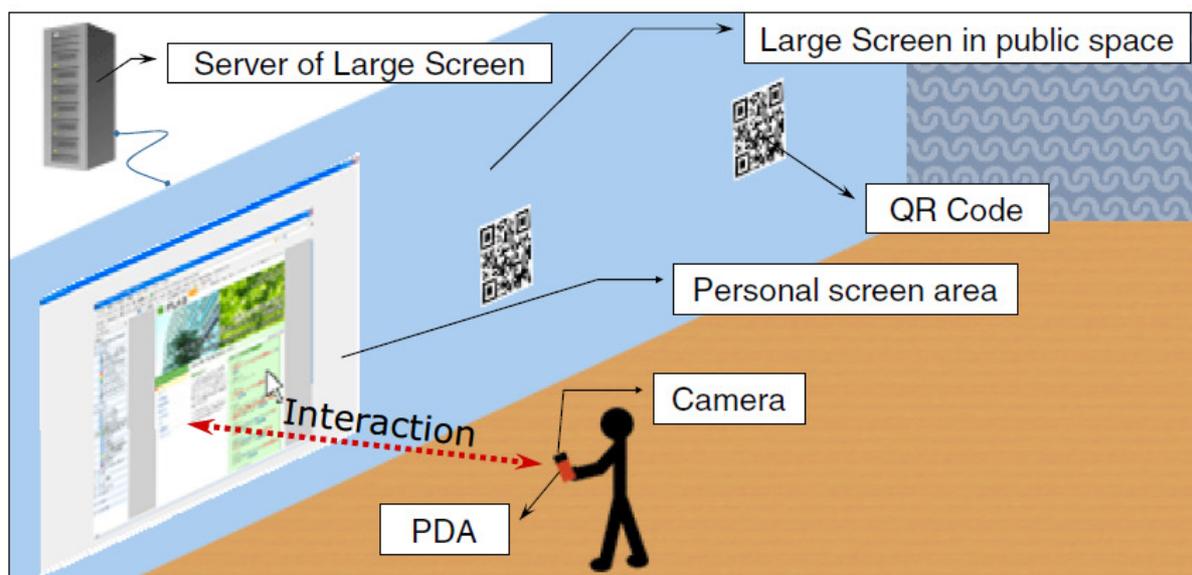


図 3-1 システム概要

図 3-1 に示したように大画面には画面表示を処理するサーバが設置されている。そして、大画面の画面には QR コードが幾つ表示されてある。ユーザはそれらの QR コードから目の前にある QR コード一つを用いて大画面と接続を行う。大画面とユーザの携帯デバイスはワイヤレスネットワークを通じて接続される。大画面とユーザの携帯デバイスとの接続が完了すると、大画面はユーザの目の前にユーザの個人領域を表示する。ユーザは個人領域に自分のデバイスにある情報を表示し、操作することができる。さらに、簡単に大画面と情報のやり取りをすることができる。例えば、大画面の情報を個人領域にドラッグしてくると、その情報がユーザのデバイスに転送され、ユーザのデバイスに保存することができる。また逆に、個人領域の情報を大画面にドラッグしていくと、ユーザのデバイスにある情報が大画面に転

送され、大画面にその情報を公開することができる。また、ユーザは個人領域を用いて、大画面で提供するいろんな情報（例えば、大画面が電車の駅に設置された場合、電車の路線図、時刻表や駅周辺の地図などの情報）、あるいはインターネットにアクセスし、インターネット上の情報を閲覧、又は操作することができる。

本研究では大画面が公共空間にあることを想定しているため、複数のユーザが同時にこのシステムを使用する場合を考慮する必要がある。本システムはマルチユーザを支援するように設計し、実行しているため、複数のユーザが同時に大画面とインタラクションをすることが可能である。

3.1 QRコードを用いた大画面と携帯デバイスの接続方法

ワイヤレスネットワークを通じて大画面と携帯デバイスを接続するためには、接続情報をユーザに提供する必要がある。本研究では IP アドレスやポート番号などのサーバ情報とユーザが使いたい大画面での位置を接続情報とする。大画面の一部分を個人領域としてユーザに提供するので大画面での個人領域の位置情報が必要となる。これらの情報を QR コードに埋め込み、QR コードを個人領域の位置に対応した位置に表示する。こうすると、QR コードの位置が個人領域を表示する位置なので、ユーザは直感的に大画面での利用したい部分を選択することができる。

QR コードとは二次元バーコードの一種であり、埋め込める情報の量が多く、読み取る方法が簡単で、読み取る速度が早いので特に日本で広く使われている。さらに QR コードを用いることで個人領域の位置情報を直感的にユーザに伝えることができる。大画面で QR コードを表示する位置を自由に指定できるので、QR コードを個人領域の位置に対応した位置に表示すれば、QR コードの位置が個人領域の位置となる。よって、ユーザは利用したい大画面の部分にある QR コードを利用すればよい。

ユーザが大画面での利用する部分を決めると、携帯デバイスにインストールされたプログラムを起動し、利用したい部分に表示してある QR コードを撮影すると、プログラムは自動的に QR コードを読み取って接続情報を取得し、大画面に大画面でのユーザが使いたい部分の位置情報と携帯デバイスの IP アドレスとポート番号を送り返す。ユーザの携帯デバイスの IP アドレスはユーザを区別するために使い、ポート番号は携帯デバイスからの情報を受け取るために使う。

QR コードを利用してユーザに接続情報を提供すると、ユーザが大画面と接続するための操作は携帯デバイスにインストールされたプログラムを立ち上げ、QR コードを撮影するだけである。この方法は操作が非常に簡単で、さらに、接続が完了するとすぐ、大画面に個人領域が表示されるので、ユーザにすぐ操作の結果を確かめるようなフィードバックが与えられる。よって、ユーザのシステムを利用する上のストレスがない。



図 3-2 大画面での QR コードの配置

図 3-2 は大画面での QR コードと大画面上の情報を表示した画面である。

3.2 個人領域

大画面と携帯デバイスとの接続が完了すると大画面に個人領域（図 3-3）を表示する。個人領域は大画面の一部分で、システムがユーザに提供するユーザの個人領域である。個人領域を利用して、ユーザは携帯デバイスにある情報を閲覧することもできれば、大画面にある情報、またはインターネット上の情報を閲覧、又は操作することができる。さらに、個人領域を通じて大画面と携帯デバイスとの情報のやり取りをすることもできる。



図 3-3 大画面に表示された個人領域

個人領域は大画面と携帯デバイスとの間の架け橋のような存在であり、大画面と携帯デバイスの間の情報のやり取りをより直感的で分かりやすくする。ユーザは個人領域を仮想の携帯デバイスだと思えば良いので、簡単に大画面に表示された情報を個人領域にドラックしてすることで、大画面の情報を携帯デバイスに保存することができれば、個人領域の情報を大画面にドラックしていくことで、携帯デバイスの情報を大画面に表示することもできる。

個人領域を通じてユーザの携帯デバイスの情報を見る（図 3-4）こともできるので、個人領域が表示された時点で、ユーザは携帯デバイスの画面を見なくてよい。それにより、ユーザの視線が大画面に固定し、大画面の画面を見たり、携帯デバイスの画面を見たりする視線の大量の移動がなくなる。それにより、ユーザはもっと集中して情報を閲覧することができる。

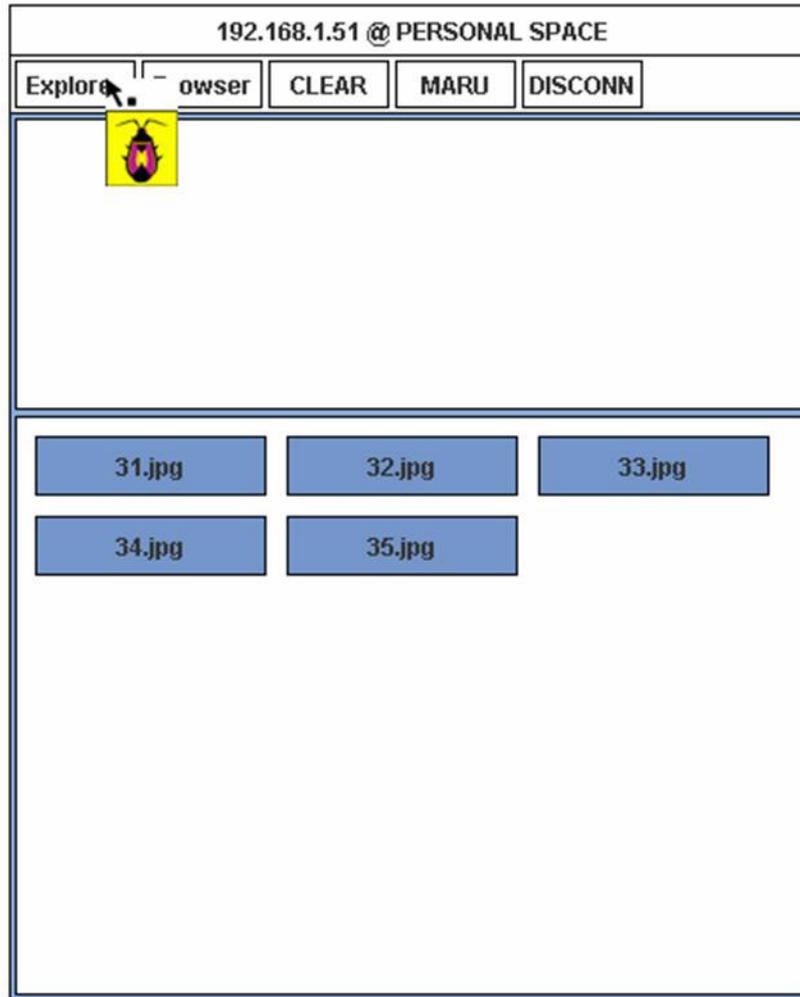


図 3-4 個人領域に表示された携帯デバイス内の情報

本研究では、携帯デバイス内の情報を大画面に表示するために、**Explorer** を開発した。**Explorer** は二つの領域に分けられている。下の領域はユーザの携帯デバイスの情報を表示する。そして、上の領域は個人領域のバッファにある情報を提示する。この部分については、3.3.2 の **MARU** インタフェースで詳しく説明する。

ユーザが大画面に個人領域を出した時点で、大画面には二種類の領域 (図 3-5) ができる。一つはユーザに提供した個人領域である。もう一つはすべてのユーザが使うことができる公共領域である。前者は指定したユーザだけに操作の権限があって、他のユーザはその領域で操作する事ができない。しかし、後者はすべてのユーザが操作する権限がある。そして、この二種類の領域の使い方も違う。個人領域は一人のユーザが使っているため、そのユーザが欲しい情報だけを表示するが、公共領域ではすべてのユーザに大画面にあるすべての情報を提示する。この二種類の領域はお互いに情報をやり取りすることはできる。

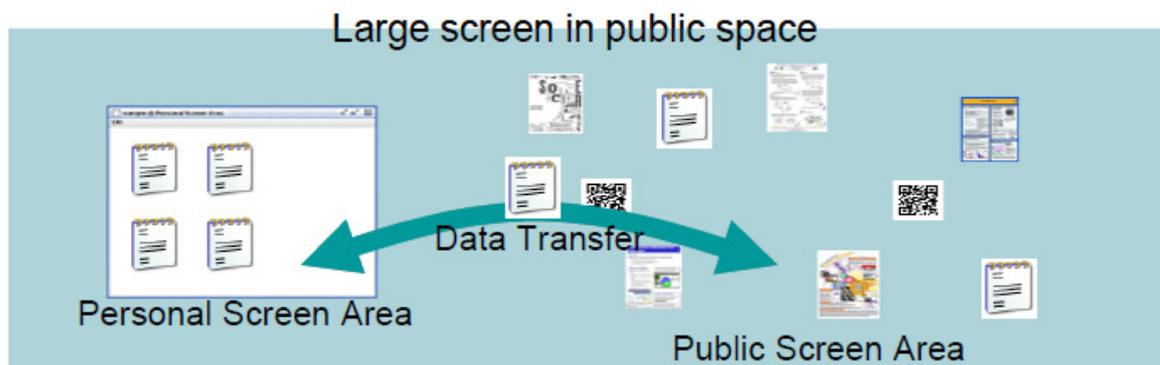


図 3-5 大画面と個人領域間の情報やり取り

あるユーザが自分の情報を大画面に公開すると、その情報が大画面の公共領域にあるとき、その情報を編集したり、削除したりする権限は公開したユーザ以外には誰にもない。しかし、もし、他のユーザがその情報を自分の個人領域に持ってきた場合は自分の個人領域内で、その情報を任意に編集したり、削除したりすることができるようになる。但し、公共領域から情報を自分の個人領域に持ってくるとしても、その情報を個人領域にコピーしてくるので、公共領域にある元の情報には影響が与えられない。

公共領域には QR コードと大画面にある情報をランダムな位置に表示する。そして、すべての情報と QR コードは重ならないように配置される。もし、ユーザが使いたい部分に情報があったとしたら、それらの情報を他の位置に移動して、個人領域に隠されたり、個人領域の上に表示されたりしないようにする。

3.3 操作インタフェース

本研究では二種類の操作インタフェースを設計した。一つは携帯デバイス側の操作インタフェースで、Personal Screen モードインタフェース（以下 PS モードインタフェース）と Common Screen モードインタフェース（以下 CS モードインタフェース）がある。もう一つは大画面側の操作インタフェースで MARU インタフェースがある。ユーザはこれらの操作インタフェースを組み合わせることで心地良く、操作をすることができる。

3.3.1 携帯デバイス側の操作インタフェース

携帯デバイスを大画面の入力装置として用いる時に、ユーザの視線の移動を考慮する必要がある。ユーザの視線の大量の移動があると、大画面の情報取得や操作に集中できなくなる。なので、携帯デバイス側の操作インタフェースはできるだけ、ユーザの視線の移動量を少なくするために、携帯デバイスが大画面と接続されている時は、携帯デバイスの画面に何の情報も提示しない方針を取っている。

本研究では携帯デバイスの画面での操作で大画面の操作を行う手法を使用する。つまり、

携帯デバイスの画面での操作で大画面でのポインティングを行う。そして、大画面にユーザ毎にユーザ専用の仮想カーソルを表示し、仮想カーソルでユーザに操作のフィードバックを与える。さらに、仮想カーソル達にイメージを付け（図 3-6）、ユーザが自分のカーソルを識別できるようにする。こうすることで、複数の仮想カーソルがある領域に集まったときでも、ユーザがどのカーソルが自分のカーソルなのかを見分けることができるようになる。



図 3-6 違うイメージを持った仮想カーソル

大画面での操作は基本的に二種類に分けることができる。一つは、個人領域内や個人領域周辺の小さい範囲での操作が考えられる。つまり、仮想カーソルが小さい移動範囲内で操作をする場合である。もう一つは、個人領域に限らず、大画面全体の広い範囲での操作が考えられる。つまり、仮想カーソルが広い範囲で操作をする場合である。

この二種類の操作を考慮して、本研究では PS モードインタフェースと CS モードインタフェースを携帯デバイス側の操作インタフェースとして設計し、実装を行った。

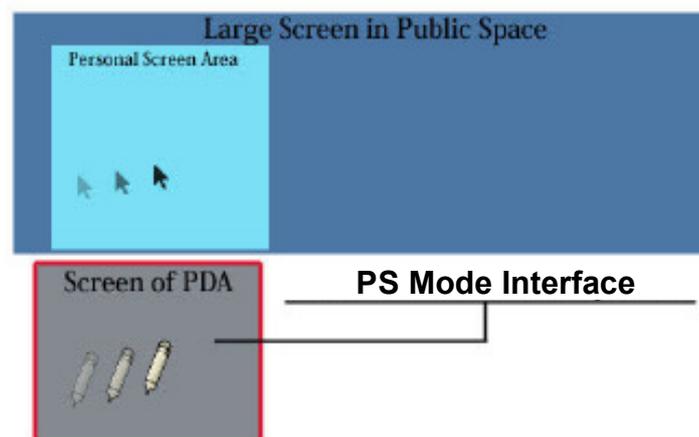


図 3-7 PS モードインタフェースの動き方

PS モードインタフェースは図 3-7 に示したようになっていて、使い方はラップトップのタッチパッドの使い方と似たような使い方をする。

PS モードインタフェースの使い方：

- 大画面の仮想カーソルを移動させるには、PS モードインタフェース上をそっとスライドすればよい。
- 大画面でクリックイベントを発生させるには、PS モードインタフェース上を軽く1回たたけばよい。
- 大画面でオブジェクトをコピーするには、仮想カーソルをオブジェクトの上に移動させ、1回たたく。2回目はPS モードインタフェース上に留め、離さず、そのままPS モードインタフェースの表面をスライドすると、仮想カーソルのイメージが変わる。そのときからはPS モードインタフェースから離してもよい。そして、仮想カーソルをドラックの目的地まで移動させる。それから軽く1回たたくとオブジェクトが目的地にコピーされる。

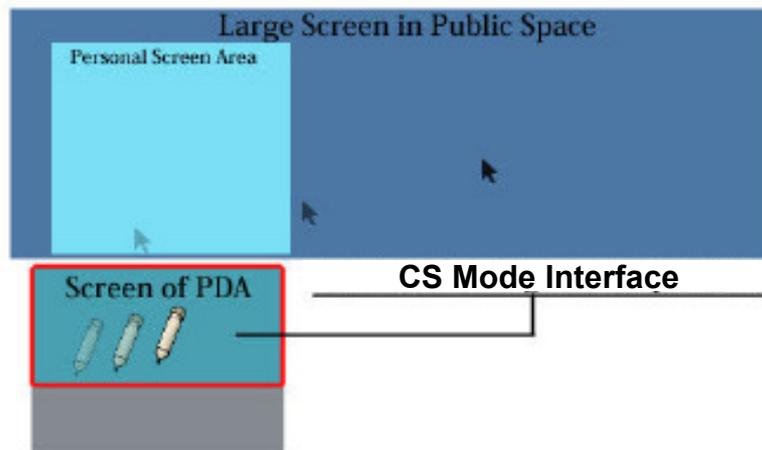


図 3-8 CS モードインタフェースの動き方

CS モードインタフェースは図 3-8 に示したようになっていて、使い方はタブレット PC のタッチパネルの使い方と同様である。しかし、CS モードインタフェースでは仮想カーソルの移動することだけができる。クリックなどのイベントの発生できない。

CS モードインタフェースの使い方：

- 大画面の仮想カーソルを移動させるには、CS モードインタフェース上をスライドすればよい。あるいは、移動させたい領域を1回たたけばよい。

3.3.2 大画面側の操作インタフェース：MARU インタフェース

本研究で提案したシステムが利用されると、大画面に大量の情報が提示されると考えられる。ユーザが大量の情報から、自分にとって有用な情報を見つけるのが困難であると考えられる。そこで、ユーザが自分にとって有用な情報を見つけることを支援するために、大画面側の操作インタフェース：MARU インタフェースを設計し、実装を行った。(MARU とは日本語のマルのローマ字での表現で、MARU インタフェースが丸くなっているのも、MARU という名前を使う。)

MARU インタフェースのコンセプト

MARU インタフェースの基本的なアイデアは Tracking Menus[15]から取ってきた。このアイデアを図 3-9 で説明する。図 3-9 の(a)に示したように、ジャーのふたの中にペンを入れた状態でペンをスライドするとする。但し、ペンはふたの底面に接触していないと仮定する。(b)のようにペンをふたの中で移動してもふたは移動されなく、元の位置にある。しかし、(c)のようにペンをふたの外に移動しようとしてふたの辺に接触した状態でスライドすると、ふたもペンの動きに従って移動される。つまり、ペンがどの位置にあっても、ふたの中へことは変らない。言い換えると、ペンがどこまで移動させても、ふたはペンに従って移動される。

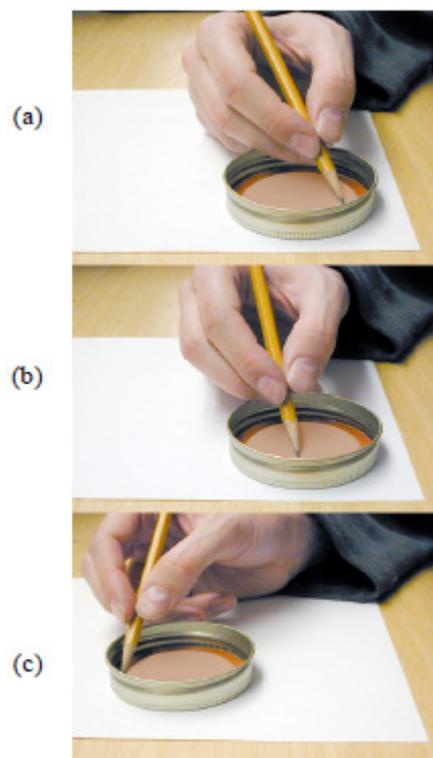


図 3-9 ジャーのふたとペンの動きの関係

このような動きには二つの特徴がある。一つは、ペンがふたの中にある時、つまり、ペンがふたの領域の中にある時は、ペンが移動しても、ふたは移動しない。もう一つの特徴はペンがふたの領域から抜け出す動きをする時、つまり、ペンがある時点でのふたの領域から外

に移動する時には、ふたはペンの移動に従い、ペンの移動する方向にペンと一緒に移動する。これらの特徴を利用すると、ふたに何らかの操作をすることができるし、さらに特別な操作をせず、自然にふたを好きな場所に移動させることができる。(図 3-9 は関連研究[15]の 72 ページにある画像である。)

MARU インタフェースはこの二つの特徴を利用して実装を行った。

MARU インタフェースの概要

MARU インタフェースは図 3-10 に示したように、マル、マルディスク、マルドーナツの三つの部分から構成されている。マルを MARU インタフェースのコンセプトでジャーのふたというメタファーを用いて説明した。そして、ペンのメタファーの役割は仮想カーソルが果たす。

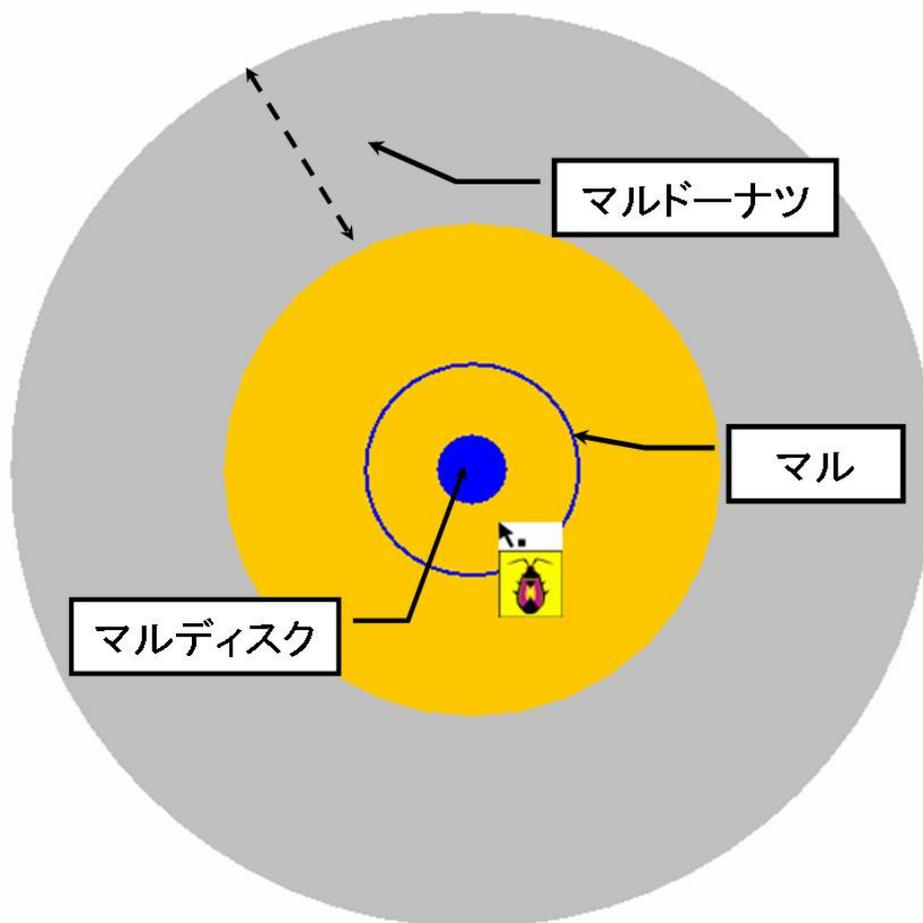


図 3-10 MARU インタフェース

図 3-10 では MARU インタフェースおよび、その構成パーツをあらわしている。

MARU インタフェースは仮想カーソルの移動に従って移動する。それから、図 3-11 のようにマルドーナツの領域内にある情報をアイコンの形でマル上に表示し、元情報をマル上表

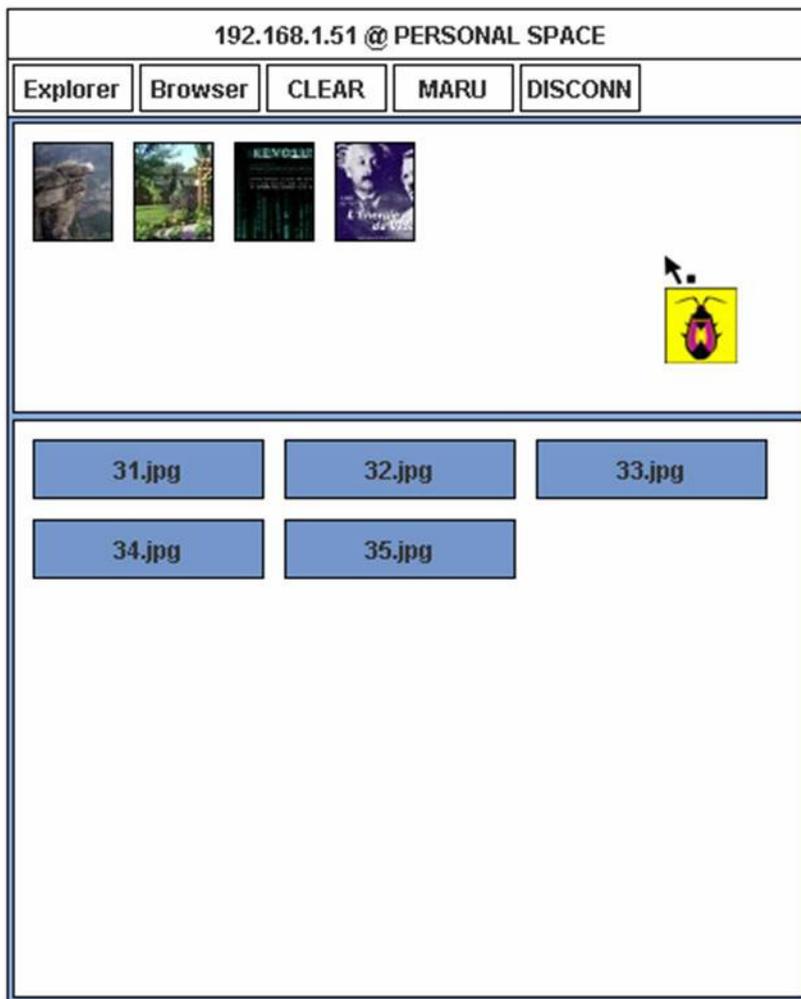


図 3-12 MARU インタフェースで収集した情報を Explorer に表示する

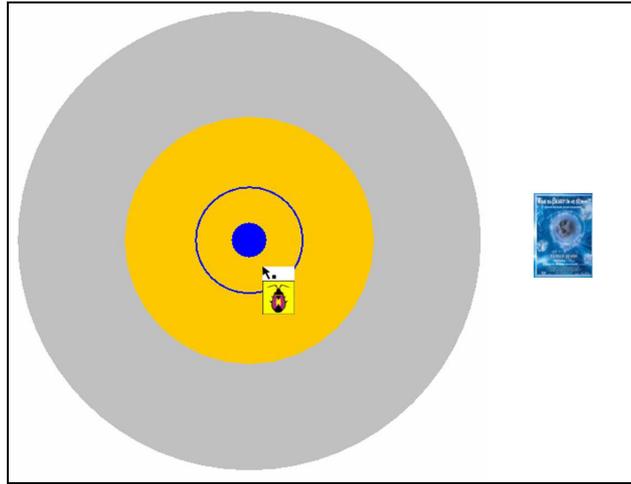


図 3-13 MARU インタフェースの情報収集方法 (a)

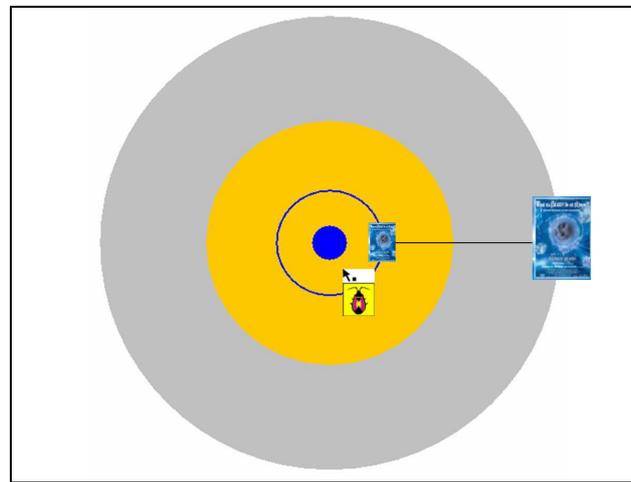


図 3-14 MARU インタフェースの情報収集方法 (b)

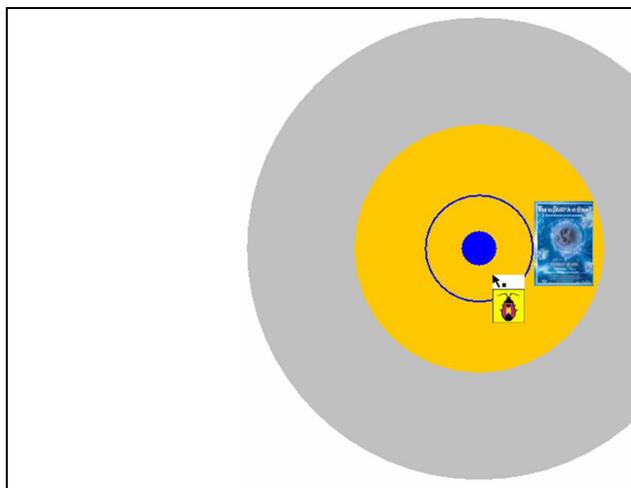


図 3-15 MARU インタフェースの情報収集方法 (c)

図 3-13 は情報が MARU インタフェースのマルドーナツの外にいる時である。図 3-14 はユーザが MARU インタフェースを移動させ、情報が MARU インタフェースのマルドーナツの上にある時である。この時は情報をアイコンの形で MARU インタフェースのマル上に表示し、アイコンと元情報を線で結ぶ。図 3-15 はユーザがさらに MARU インタフェースを移動させ、情報が MARU インタフェースの内側になる時である。この時は MARU インタフェースのマルからアイコンを表示しない。

本研究で提案したシステムはマルチユーザシステムである。従って、ユーザ毎に MARU インタフェースを用意する必要がある。本システムでは、ユーザ毎に MARU インタフェースが用意され、これらの MARU インタフェースは独立していて、お互いに影響を与えない。つまり、あるユーザが他のユーザの MARU インタフェースに対して操作することはできない。そして、ある MARU インタフェースのマルドーナツは他の MARU インタフェースのすべての領域を重ねることができる。しかし、二つの MARU インタフェースのマル領域は重ねることができない。このような仕組みを採用すると、すべてのユーザが大画面上のすべての情報を収集することができるが、お互いの操作を邪魔することがなくなる (図 3-16)。



図 3-16 二つの MARU インタフェースが隣接した場合

サーバ側では最初に情報管理部からサーバ上にある情報と QR コードを画面に表示し、ユーザの接続を待つ。

ユーザが QR コードを撮ると、接続部で QR コードを解析し、接続情報を読み取り、その接続情報に基づいて、サーバ側の接続管理部に接続情報を送る。

接続管理部では接続情報を受け取って、接続情報に基づいてユーザ情報を作り、ユーザ情報を Client 管理部に渡す。Client 管理部では、ユーザ情報を利用し、新しく Client を生成する。新たな Client が生成されるとともに、Client は個人領域、仮想カーソルと MARU インタフェースが生成し、接続情報にあった位置に個人領域と仮想カーソルが配置される。個人領域は、まずユーザ側からユーザのデータを取って来る。ユーザの操作によって、取ってきたユーザのデータを提示したり、インターネット上の情報を提示したりすることができる。仮想カーソルはユーザの操作を実行する。MARU インタフェースは大画面での情報閲覧に使用される。

ユーザが操作をし始めるとユーザ側の操作インタフェースでその操作情報をサーバ側の操作受付に送る。サーバ側の操作受付では操作を仮想カーソルに渡す。仮想カーソルはこれらの操作を個人領域や情報管理部に対して実行させる。

4.2 サーバ側の実装

サーバ側のプログラムを起動すると、メインフレームが呼び出される。メインフレームはまず、情報提示部と情報管理部を生成し、情報管理部によって QR コードとサーバ上のすべての画像データをロードし、情報定時部で重ならないように位置を指定する。そして、すべての画像データの領域を登録しておく。これは操作する時、画像データを選択する際に使用する。それから、メインフレームは仮想カーソルのイメージをロードし、保存する。

メインフレームがこれらの作業を終えると、接続管理部を生成する。接続管理部はソケットサーバを作り、ユーザからの接続を待つ。

ユーザが接続を図ると、ユーザ側の接続部から接続情報が接続管理部に送られて来る。接続情報は文字列の形であって各情報が『:』で区切られている。接続情報の例を挙げると『10.10.10.1:7000:300.200』がある。最初の『10.10.10.1』がユーザ側の IP アドレスであり、『7000』は操作を受け取るポート番号、それから『300:200』は個人領域を表示する座標である。接続管理部が接続情報を受け取ると『:』で区切ってこれらの情報を Client 管理部に渡し、Client 管理部から新しい Client を生成する。生成された Client はまた、操作を受け取るポート番号と位置情報を用いて仮想カーソルを生成し、ユーザ側の IP アドレスと操作を受け取るポート番号に 10 を足したポート番号、データを交換する際に使用するポート番号と位置情報を用いて個人領域を生成する。



図 4-2 大画面と携帯デバイスの接続を行う様子

個人領域は生成される際、接続情報の位置情報によって大画面での位置が決められる。それから、個人領域はまた、データを受け取るポート番号を使用して **Explorer** を生成する。**Explorer** は 4.2.1 の個人領域の実装の方で詳しく説明する。

仮想カーソルが生成されると、まず、メインフレームから使っていないカーソルイメージを取ってきて、接続情報の位置情報に基づいて画面に配置する。そして、接続情報の操作を受け取るポート番号で **UDP** ソケットサーバを作り、ユーザ側からの操作を待つ。これで、サーバ側とユーザ側の接続が完了する。図 4-2 はユーザが大画面と携帯デバイスを接続するため、**QR** コードを撮っている様子である。

4.2.1 個人領域の実装

表示された個人領域には五つのボタンがあり、その下には何の情報も表示されていない。ボタンの機能はこのようになっている。**Explorer** ボタンは **Explorer** を開く。**Browser** ボタンはブラウザを開き、インターネットのウェブサイトを見ることができる。**Clear** ボタンは **Explorer** と **Browser** を閉じる。**MARU** ボタンは個人領域を非表示にし、**MARU** に切り替える。**Disconnect** ボタンはユーザ側とサーバ側の接続を切る。

ユーザの操作により、**Explorer** ボタンが押された時、個人領域は **Explorer** を表示する。**Explorer** は生成される際に、ユーザ側からユーザのデータを取って来て、**Explorer** の下の枠に表示する。**Explorer** の上の枠には個人領域のバッファに保存された画像データを表示する。**Explorer** には **Commander** があって、**Commander** でサーバ側とユーザ側のデータのやり取りを行う。

ユーザ側にデータリストを送るという命令を送信し、ソケットサーバを立ち上げ、ユーザ側からのデータリストを受信する。同じように指定したデータの受信も行う。データを送る時にはデータを受信するという命令を送信し、データをユーザ側に発信する。このことによりサーバ側とユーザ側のデータのやり取りを実現した。

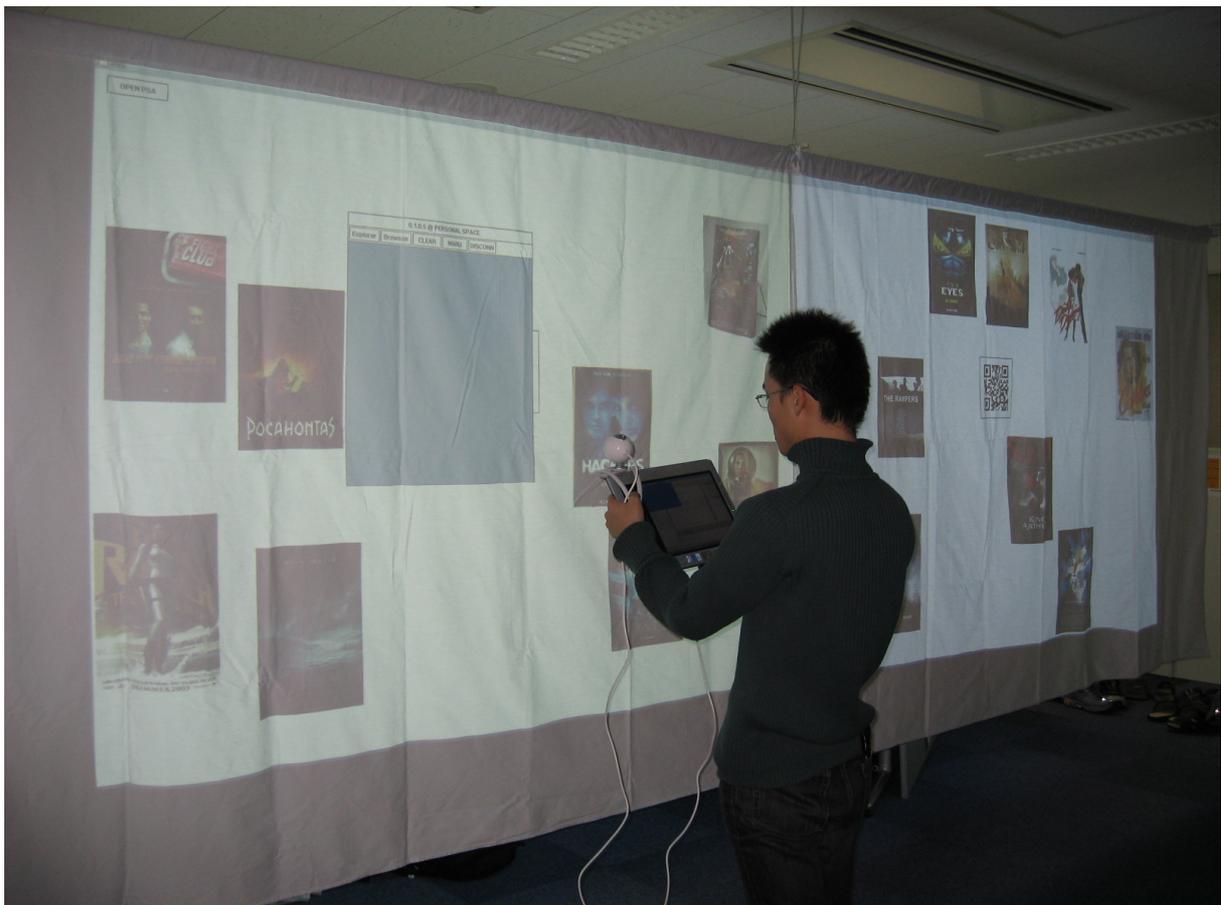


図 4-3 大画面で個人領域を出した様子

図 4-3 はユーザが大画面と携帯デバイスの接続を完了し、大画面にユーザの個人領域を表示した様子である。

4.2.2 操作の実行の実装

大画面での操作はすべて仮想カーソルを通じて実行される。仮想カーソルが生成されると UDP ソケットサーバを立ち上げ、ユーザ側からの操作メッセージを待つことになる。操作メッセージは PS モードインタフェースから作られる **Personal Screen** モード (以下 PS モード) と、CS モードインタフェースから作られる **Common Screen** モード (以下 CS モード)

がある。それぞれの操作メッセージのフォーマットは以下のようになっている。

PS モード :

メッセージモード : ユーザの IP : ユーザの port : X 軸の移動量 : Y 軸の移動量 : イベントタイプ

CS モード :

メッセージモード : ユーザの IP : ユーザの port : X 軸の位置 : Y 軸の位置 : 携帯デバイスの X 軸方向の長さ : 携帯デバイスの Y 軸方向の長さ : イベントタイプ

例)

PS モード : 1:10.10.10.1:10:2:1

CS モード : 2:10.10.10.1:230:134:300:400:8

サーバ側の操作受付でこれらの操作メッセージを受け取り、イベントに変換し、仮想カーソルのイベントキューに入れる。仮想カーソルは定期的にイベントキューをチェックし、イベントをとりだして、そのイベントを実行する操作の実行の主なことは仮想カーソルの移動とオブジェクトドラッグイベントである。

仮想カーソルの移動

仮想カーソルの移動は操作メッセージのモードにより異なる。

PS モード (図 4-4) の時は、予め仮想カーソルが生成されるときに位置をバッファに保存しておく。新しくイベントが実行される時にはバッファにある位置にイベントの移動量を足して、今回のカーソルの位置を求める。そして、その位置に仮想カーソルを配置し、また、その位置をバッファに保存する。このような操作を繰り返して仮想カーソルの移動を実現する。

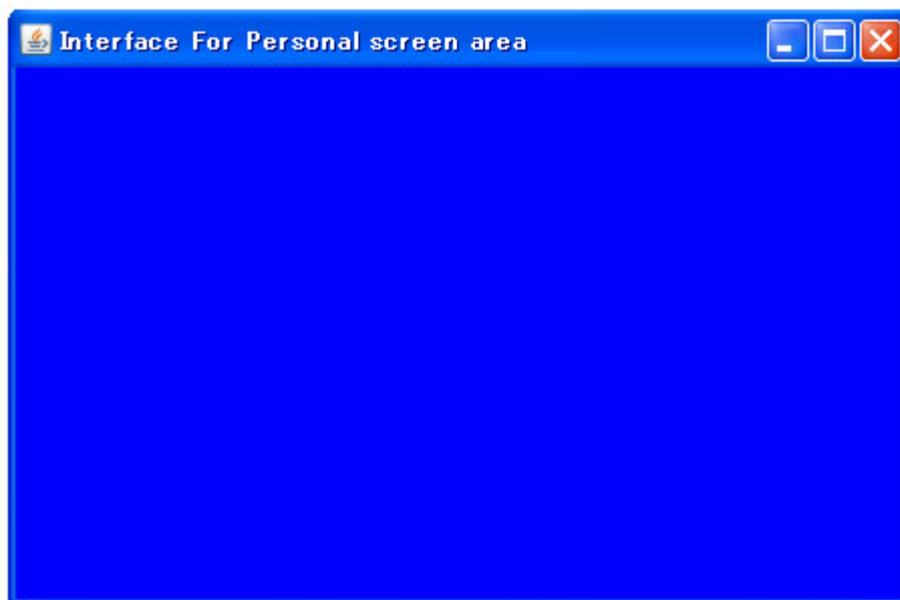


図 4-4 PS モードインタフェース

図 4-4 は PS モードインタフェースが携帯デバイスの画面に表示された時である。

CS モード (図 4-5) の時は、携帯デバイスのサイズ、携帯デバイスでのカーソルの位置と大画面のサイズによって、大画面での仮想カーソルの位置を求めて、仮想カーソルを配置する。

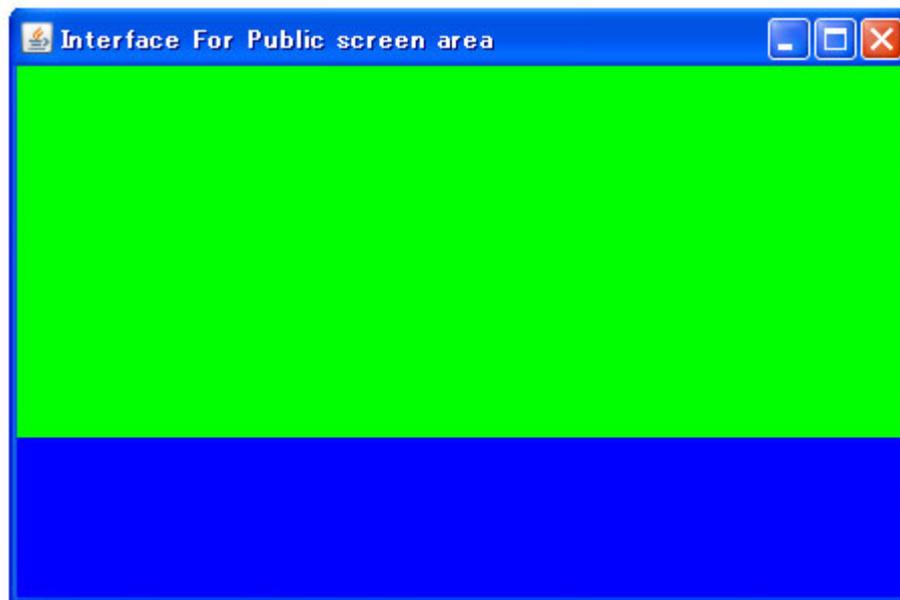


図 4-5 CS モードインタフェース

図 4-5 は CS モードインタフェースが携帯デバイスの画面に表示された時である。

オブジェクトドラックイベント

ドラックイベントを実現するためにはまず、ドラックの開始とドラックの終了を判断する必要がある。今回の実装ではイベントのバターンマッチングでドラックの開始と終了を判断することにした。仮想カーソルで前 6 回のイベントのイベントタイプをバッファに保存する。そして、下のバターンをバターンマッチングして、あつてれば、ドラックの開始、あるいはドラックの終了とみなす。

ドラック開始バターン：

- 1 Mouse pressed
- 2 Mouse released
- 3 Mouse clicked
- 4 Mouse pressed
- 5 Mouse moved

ドラック終了バターン：

1. Mouse moved
2. Mouse moved
3. Mouse released
4. Mouse pressed
5. Mouse released
6. Mouse clicked

ドラッグ開始のイベントが発生した場合、その位置によって画像データを取得し、仮想カーソルのイメージをドラッグ時のイメージに変え(図 4-6)、ドラッグの状態に入る。ただし、MARU インタフェースで操作をする時には位置によって画像データを取得するのではなく、mouse clicked イベントにより画像データを取得する。

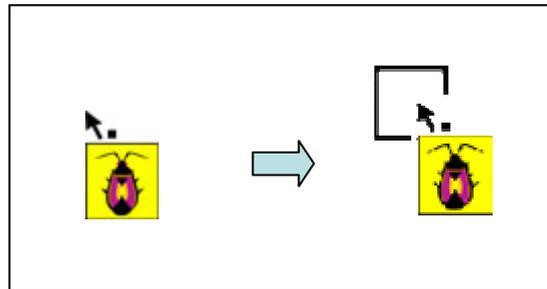


図 4-6 ドラッグイベントが発生時、仮想カーソルのイメージを変える

その次にドラッグ終了のイベントが発生した場合、まず、ドラッグの状態であるかをチェックする。ドラッグ状態であれば仮想カーソルが個人領域の領域内にあるかを判断し、領域内であれば個人領域に画像データを渡す。個人領域でバッファ空間に表示するか、ユーザ側に送るかをドラッグ終了イベントが発生した位置によって判断し、実行される。

4.2.3 MARU インタフェースの実装

サーバ側で Client が生成されると、Client がまた MARU インタフェースを生成する。MARU は円形で内側に塗りつぶされたマルディスクがあり、外側にドーナツ状のマルドーナツがある。仮想カーソルをマルディスクの上に移動するとマルディスクの色がかわる。仮想カーソルが画像データをドラッグ状態で、マルディスクでドラッグ終了イベントが発生させるとドラッグされた画像データが個人領域のバッファ空間に伝送される。仮想カーソルがドラッグ状態でないときにマルディスクでドラッグ終了イベントが発生すると MARU を非表示にし、個人領域に切り替える。

画像データのアイコンの表示

MARU インタフェースは生成されると情報管理部から大画面に表示されたすべての画像データの情報を取ってくる。そして、MARU インタフェースの中心点から各画像データまでの距離を計算する。その距離が画像データの半径(実装の便利上、四角の画像データの短い変の半分を画像データの半径とする)と MARU インタフェースのマルドーナツ外半径(図 4-5)の和より小さい、且つ、画像データの半径と MARU インタフェースのマルドーナツ内半径(図 4-5)より大きい時、その画像データのアイコンを作り、マルの上に表示させる。さらに、アイコンの中心からその画像データの中心までを線で繋ぐ。

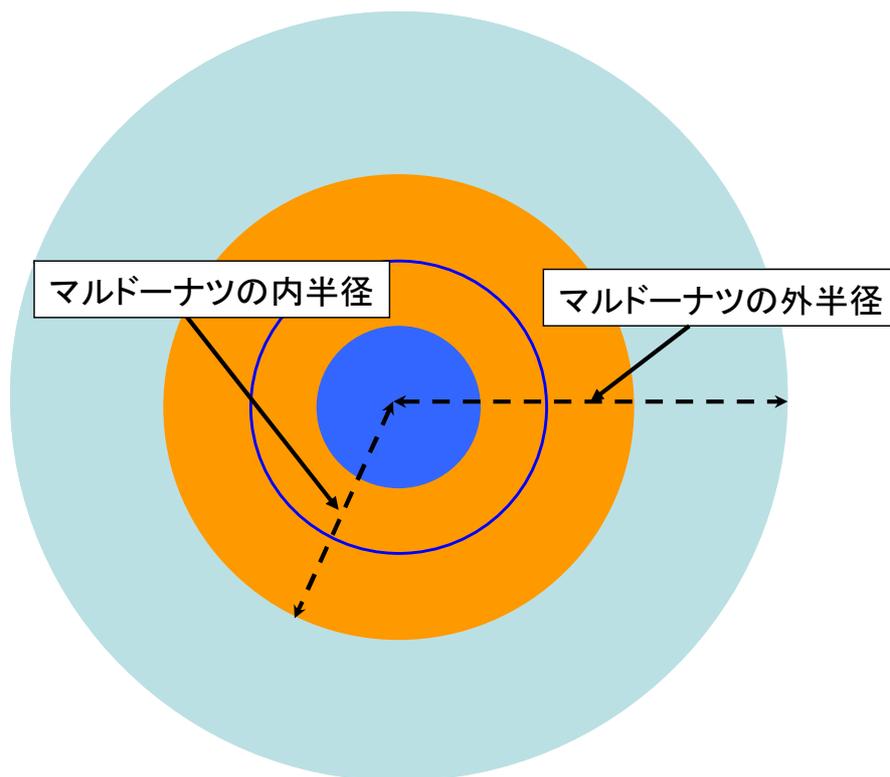


図 4-7 マルドーナツとマルディスクの半径

図 4-7 は MARU インタフェースのパーツであるマルドーナツの内半径をマルドーナツの外半径を表している。

MARU インタフェースの移動

仮想カーソルがイベントキューからイベントを取り出す時、イベントの位置情報を Client に保存しておく。さらに、ユーザが位置情報を保存する時に MARU インタフェースもその位置情報を渡す。仮想カーソルが MARU インタフェースのマルから出ようとしたら、MARU インタフェースを仮想カーソルと同じ方向で、同じ移動量ほど移動させる (図 4-8)。

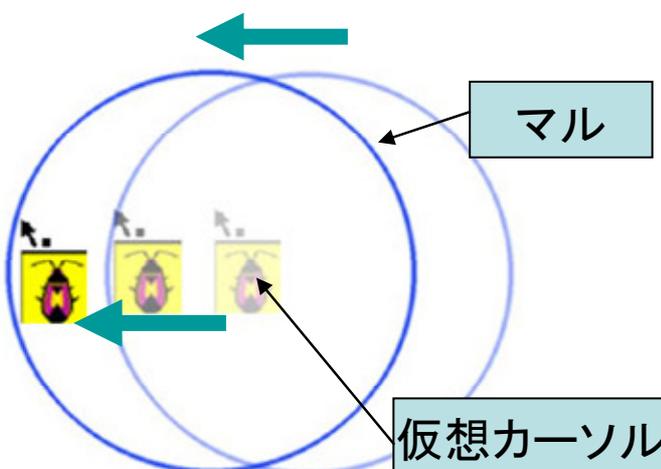


図 4-8 MARU インタフェースの移動

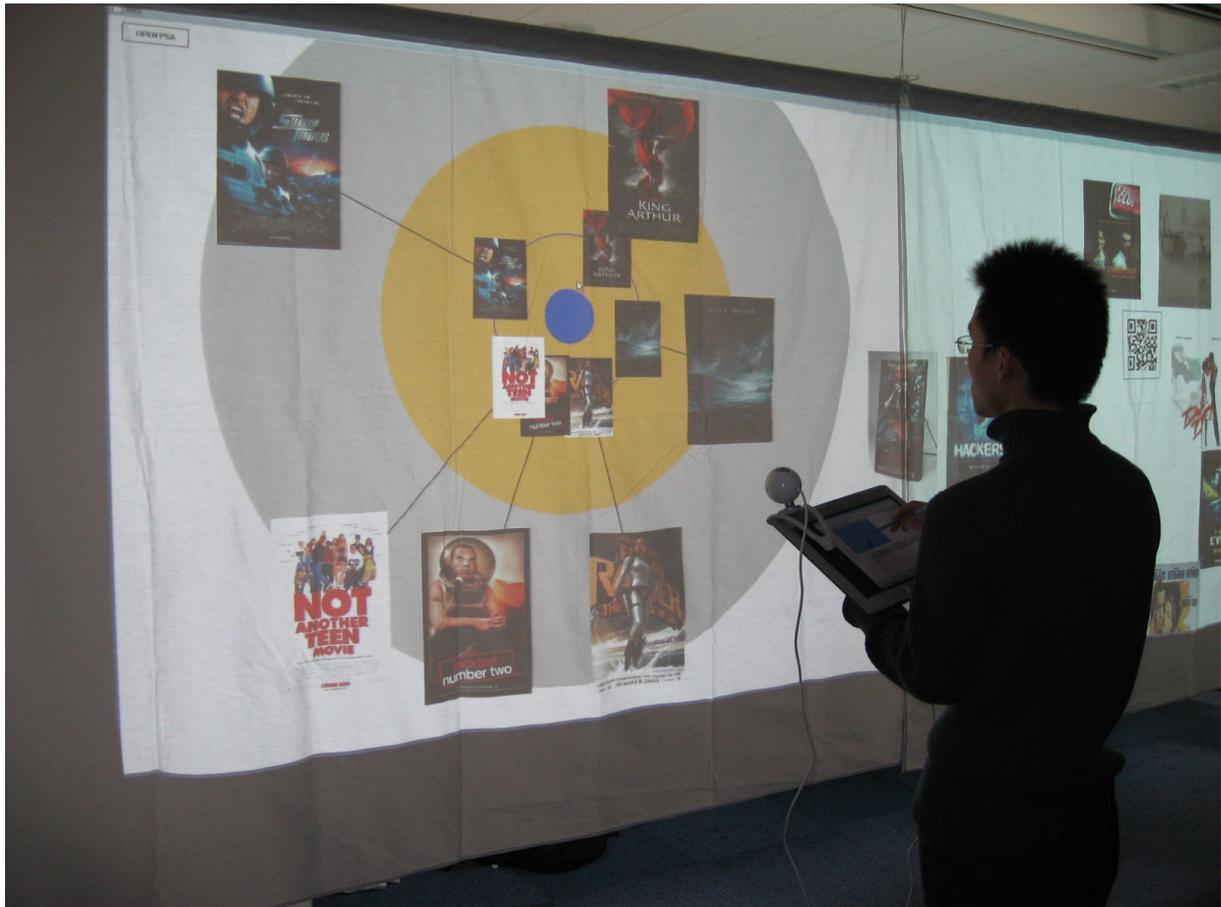


図 4-9 MARU インタフェースを利用している様子

図 4-9 はユーザが MARU インタフェースを利用して、大画面に提示された情報を収集している様子である。

4.3 ユーザ側の実装

ユーザ側の接続部で QR コードを解析するために `qrcode` ライブラリ[注]を利用した。QR コードの撮影には USB ウェブカメラを使用した。USB ウェブカメラを使用するために `Java Media Framework2.1.1e` を利用した。

ユーザ側のプログラムを起動すると、20 ミリ秒間隔で USB カメラから画像を取って来て、ユーザの画面に表示する。そして、1 秒間隔で画像を `qrcode` ライブラリに送り、QR コードの解析を行う。`qrcode` ライブラリから帰ってくる結果は三種類ある。一つは解析失敗で値が `null` である。二つ目は解析ができて値が `null` ではないが、解析が間違っていて、正確な値ではない。三つ目は解析が成功し、正確な値が帰ってくる。間違った解析結果を使用することを防ぐために、`qrcode` ライブラリから帰ってきた値は解析が正確に行ったのかをチェックする必要がある。もし、値が `null` や間違った解析結果であれば、正確の値が来るまでずっと解析を行う。

注 <http://sourceforge.jp/projects/qrcode/>

正確な解析結果が帰ってきた時は、USB カメラを止めて、次の処理に入る。解析結果の IP アドレスだけをユーザのアドレスに書き換えて、サーバ側の接続管理部に送る。接続情報を送ることに成功すれば、PS モードのインタフェースを開き、さらに、QR コードを解析した結果の IP アドレスとポート番号でソケットサーバを作り、サーバ側からの指示を待つ。データリストの要求があるならば、指定したディレクトリのデータのリストを作り、サーバに送る。送り終わるとまた、指示待ち状態に入る。このような仕組みでサーバにデータリストやデータを送ったり、サーバ側からデータを受け取ったりする。

ユーザが PS モードインタフェースに操作をすると、インタフェースはカーソルの移動量を計算し、イベントタイプなどの情報を用いて、操作メッセージを作り、サーバ側の操作受付に送る。サーバ側の操作受付のポート番号は QR コードの解析で取ったポート番号を使用する。

PS モードインタフェースは CS モードインタフェースに切り替えることができる。PS モードインタフェースでクリックイベントを連続 3 回発生させると CS モードインタフェースに切り替える。CS モードインタフェースの下の青い領域を 2 回クリックすると PS モードインタフェースに切り替える。

CS モードインタフェースは大画面でのイベントの発生は行わない。ただ、大画面での仮想カーソルの移動を行う。操作メッセージの作り方は PS モードインタフェースと同様であるが、操作メッセージに CS モードインタフェースのサイズを加える。こうして、作った操作メッセージをサーバ側に送る。

4.4 iWS の使用例

このセクションでは iWS の利用例を二つほど挙げる

4.4.1 電車の駅での利用例

山田さんはラーメン屋を営んでいるが、最近お客さんが増えて、アルバイト募集をしたいとする。この場合、予めアルバイト募集のポスターをつくり、PDA に保存する。それで、出勤する時、電車を待っているうちに田町駅に設置してある大画面の前に行く。PDA を出して、目の前に表示した QR コードを撮ると、大画面に個人領域が表示され、個人領域に PDA に保存しておいたアルバイト募集ポスターが表示される。PDA の画面に操作して、大画面のカーソルを動かして、個人領域に表示されたアルバイト募集ポスターをドラッグして大画面に置く。そうすると、アルバイト募集ポスターが電車の駅に設置した大画面に公開させる。

翌日の朝、大学生の佐藤君は田町駅で電車を待ちながら、大画面を見ている。そこで、山田さんが公開したアルバイト募集ポスターを見つける。最近、ちょうどアルバイトをすることにした佐藤君は PDA を出して目の前に表示された QR コードをとり、個人領域を出す。そして、そのアルバイト募集ポスターを自分の個人領域にドラッグしてくる。そうすると、アルバイト募集ポスターが佐藤君の PDA に保存される。学校の授業が終わった後、佐藤君は PDA からアルバイト募集ポスターを見直し、条件が良さそうなのでポスターにあった電

話番号で山田さんに連絡する。

4.4.2 ショッピングモールでの利用例

井上さんは日曜日にショッピングモールにスーツを買いに行ったとする。井上さんはまず、ショッピングモールに設置されてある大画面の前に行く。大画面にはショッピングモールで売られている商品の情報が表示されている。井上さんは大画面に表示された商品が多すぎて戸惑っている。そのとき、井上さんは PDA を出して、大画面と接続し、表示された個人領域を MARU インタフェースに切り替える。そうすると、MARU インタフェースは周りの情報を取ってきてアイコンで表示する。井上さんは MARU インタフェースでちょっとでも興味を持った商品を MARU インタフェースのマルディスクにドラックしてくる。商品情報をいくつかドラックしてきた後、個人領域に切り替え、取ってきた情報を比べながら見る。それから、気に入った表品を PDA に保存する。そして、PDA と大画面の接続を終了し、PDA で商品の情報を見ながら、商品の売り場に向かう。

第5章 iWS の考察

本章では提案システム iWS の利点と今後の改善点について述べる。

5.1 提案システム iWS の利点

壁サイズ大画面と携帯デバイスを組み合わせて使うことで、壁サイズ大画面がインタラクティブになるようになった。ユーザが直接大画面の情報に対して操作することができて、大画面に提示した情報をユーザの携帯デバイスに保存したり、携帯デバイスの情報を大画面に公開したりすることができるようになった。さらに、携帯デバイスの小さい画面問題を克服し、大画面で携帯デバイスの情報を写して操作することで、大画面の広い画面で心地良く操作することができるようになった。

QR コードを用いて大画面と携帯デバイスの接続を行う手法を利用して、このような環境を利用するうえでの煩わしい接続操作を簡単にさせた。ユーザは携帯デバイスのカメラで QR コードを撮影することだけで、携帯デバイスと大画面との接続を行うことができる。それから、携帯デバイスを大画面の入力装置として利用することで、大画面での操作のための専用デバイスを用いず、簡単に大画面での操作をすることができる。

5.2 今後の改善点

5.2.1 プライバシーの問題

ユーザが携帯デバイスの情報を大画面に表示するうえにプライバシーの問題が存在する。本研究では対策として、携帯デバイスにシステム専用のフォルダーを作る。携帯デバイスが大画面と接続され、携帯デバイスの情報を大画面に表示するとき、その専用のフォルダー内の情報をすべて大画面に表示するようにした。ユーザは大画面を利用する前に、大画面に表示しても良い情報をその専用フォルダーにコピーしておけばよい。こうすると、ユーザが意外に人に見られたくない情報を大画面に表示することをある程度防ぐことができる。

しかし、本システムは携帯デバイスが大画面と接続されている間には携帯デバイスの画面を見ない方針を採っているため、大画面で操作をしながら携帯デバイスの情報を操作することができない。そのため、大画面で操作している間に大画面に出したい情報が専用フォルダーにないと、一回大画面との接続を切って、その情報を専用フォルダーにコピーし、再び大画面との接続を行う必要がある。このような作業は煩わしくて望ましくない。

Hybrid images[18]では距離によって同じ画面の画像が違うように見えるような研究を行っている。このような研究が発展すると、大画面に出した携帯デバイス内の情報がそのユーザだけが正確に見えるようになれる。そうすれば、プライバシーの問題は解決できる。

5.2.2 セキュリティ問題

ユーザが携帯デバイスの情報を大画面に表示する時に、本システムでは携帯デバイスのデータリストを大画面に送る。このような個人情報が大画面に送られるので、セキュリティ問題を起こす可能性がある。

この問題を解決するためにはVNC技術[注]を利用して個人領域の実装をすれば良い。VNC技術を利用すると、個人領域の画面を大画面で生成するのではなく、携帯デバイスの方で生成し、大画面に生成された画像情報を送り、大画面でそれを表示する。こうすると、境内デバイスから大画面に送られるのは、生成された画面の画像データだけで、他に何の情報も送らないので、セキュリティ問題を起こすことを防ぐことができる。

それから、遠隔地から大画面に接続して何らかの情報を画面に表示する恐れがある。この問題の対策としてはShownPass[7]がある。ShownPassは物理的に近くにいるユーザがソースを利用できるような仕組みである。QRコードにパスワードを追加し、ある時間間隔でパスワードを変更すると、大画面の前にいるユーザだけが画面を使えるようにすることができる。

5.2.3 大画面に公開された情報の管理

ユーザが大画面に公開した情報を管理する必要がある。情報の公開期間を決める必要がある。公開された情報を大画面から削除しないと、大画面の情報が増える一方で、いずれは新しい情報を公開する空間がなくなる。なので、公開された情報に公開期間を設定し、その期間だけに大画面に表示し、公開期間を超えると自動で画面から削除するのが良いであろう。

それからは、公開された情報の提示方法がある。ユーザが提示する情報の形がいろいろあると考えられる。例えば、テキストの情報、画像情報、音声情報や動画情報などがありえる。これらの情報を大画面でどんな形で提示するのかを考える必要がある。本システムではこれらの情報を画像情報であると仮定して実装を行ったが、実際はもっといろんな形の情報があられる。情報の提示は元の形で提示するのが一番良いと考えられる。テキストの情報はテキストで提示し、画像情報は画像で提示するのが望ましい。

今後、この問題について、もっと深く研究すべきである。

注 http://ja.wikipedia.org/wiki/Virtual_Network_Computing

第6章 おわりに

本研究では携帯デバイスを公共空間の壁サイズ大画面の入力装置として用いて、壁サイズ大画面とのインタラクションを行う手法を提案し、プロトタイプとして iWS の実装を行った。提案システムにより、ユーザが携帯デバイスを用いて大画面での操作や情報交換をすることができるようになった。さらに、QR コードを用いて大画面と携帯デバイスの接続を行う手法を利用して、大画面と携帯デバイスの接続を簡単にできるようにした。それから、大画面に提示した大量の情報の閲覧を支援する MARU インタフェースの実装を行った。

謝辞

本論文の執筆にあたり、指導教員である筑波大学システム情報工学研究科田中二郎教授から大変貴重な意見やご指導をいただきました。心から感謝致します。また、筑波大学システム情報工学研究科高橋伸講師からは、研究内容だけではなく、研究の進捗についてもご指導をいただきました。心から感謝いたします。そして、筑波大学システム情報工学研究三末和男助教授や志築文太郎先生から大変貴重な助言をいただきました。ここで深く御礼を申し上げます。筑波大学システム情報工学研究科インタラクティブプログラミング研究室の方々にも大変お世話になりました。最後に私を支えてくれた両親や、すべての友人に感謝を申し上げます。本当にありがとうございました。

参考文献

- [1] Jun Rekimoto, A multiple-device approach for supporting whiteboard-based interactions. In Proceedings of ACM CHI'98, pp. 344-351, February 1998.
- [2] Rafael Ballagas, Michael Rohs, Jennifer G. Sheridan. Sweep and point and shoot: phonecam-based interaction for large public displays, ACM CHI'05, pp.1200-1203, 2005.
- [3] Yoo-Joo Choi, Seong Joon Yoo, Soo-Mi Choi, Carsten Waldeck, Dirk Balfanz, User-Centric Multimedia Information Visualization for Mobile Devices in the Ubiquitous Environment, Knowledge-Based Intelligent Information & Engineering Systems(KES2006), October 9-11, 2006, Bournemouth, U.K. (LNAI). Vol. 4251, pp. 753-762.
- [4] Keith Cheverst, Alan Dix, Daniel Fitton, Chris Kray, Mark Rouncefield, George Saslis-Lagoudakis, Jennifer G. Sheridan. Exploring Mobile Phone Interaction with Situated Displays. PERMID workshop at Pervasive 2005, Munich, PERVASIVE 2005, pp.43-37.
- [5] Himanshu Raj, R. Gossweiler, D. Milojicic, "Content Cascade Incremental Content Exchange between Public Displays and Personal Devices", in Proc. of the first Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous'04), Boston, Massachusetts, USA, pp.374-381, 2004.
- [6] Ferscha, A. and S. Vogl, Pervasive Web Access via Public Communication Walls: Pervasive Computing, Springer LNCS 2414, Zurich, Switzerland, pp.84-97, 2002.
- [7] Yuji Ayatsuka, Michimune Kohno, Jun Rekimoto, "Real-World Oriented Access Control Method with a Displayed Password" in Computer Human Interaction (APCHI 2004), LNCS 3101, Jun. 2004, pp.9-29
- [8] C. Jin, S. Takahashi and J. Tanaka, Interaction between small size device and large screen in public space, Proceedings of the 10th International Conference on Knowledge-Based & Intelligent Information & Engineering Systems (KES2006), Part III, LNAI 4253, pp.197-204, Bournemouth, United Kingdom, October 9-11, 2006.
- [9] Maxime Collomb , Mountaz Hascoët , Patrick Baudisch , Brian Lee, Improving drag-and-drop on wall-size displays, Proceedings of the 2005 conference on Graphics interface, pp.25-32, May 09-11, 2005, Victoria, British Columbia
- [10] Shahram Izadi, Harry Brignull, Tom Rodden, Yvonne Rogers, Mia Underwood, Dynamo: A Public Interactive Surface Supporting the Cooperative Sharing and Exchange of Media. Proceedings of the 16th annual ACM symposium on User interface software and technology, pp.159-168, UIST'03, 2003.

- [11] Johanson B, Fox A, Winograd T, The Interactive Workspaces project: experiences with ubiquitous computing rooms, *Pervasive Computing*, IEEE, Volume 1, Issue 2, pp.67-74, Apr-Jun 2002.
- [12] Brad A Myers, Herb Stiel, Robert Gargiulo, Collaboration Using Multiple PDAs Connected to a PC, *Proceedings of the 1998 ACM conference on Computer supported cooperative work*, pp.285-294, 1998.
- [13] Kento Miyaoku, Suguru Higashino, Yoshinobu Tonomura, C-Blink: A Hue-Difference-Based Light Signal Marker for Large Screen Interaction via Any Mobile Terminal, *Proceedings of the 17th annual ACM symposium on User interface software and technology*, pp.147-156, 2004.
- [14] Seok-Hyung Bae, Takahiro Kobayashi, Ryugo Kijima¹, Won-Sup Kim, Tangible NURBS-curve Manipulation Techniques Using Graspable Handles on a Large Display, *Proceedings of the 17th annual ACM symposium on User interface software and technology*, pp.81-90, 2004.
- [15] George Fitzmaurice , Azam Khan , Robert Piek, Bill Buxton , Gordon Kurtenbach, Tracking menus, *Proceedings of the 16th annual ACM symposium on User interface software and technology*, p.71-79, November 02-05, 2003, Vancouver, Canada
- [16] Anastasia Bezerianos , Ravin Balakrishnan, The vacuum: facilitating the manipulation of distant objects, *Proceedings of the SIGCHI conference on Human factors in computing systems*, April 02-07, 2005, Portland, Oregon, USA
- [17] Azam Khan , George Fitzmaurice , Don Almeida , Nicolas Burtnyk , Gordon Kurtenbach, A remote control interface for large displays, *Proceedings of the 17th annual ACM symposium on User interface software and technology*, pp.127-136, October 24-27, 2004, Santa Fe, NM, USA
- [18] Aude Oliva, Antonio Torralba, Philippe. G. Schyns, Hybrid images, *International Conference on Computer Graphics and Interactive Techniques, ACM SIGGRAPH 2006 Papers*, pp.527-532, 2006.
- [19] Harry Brignull, Shahram Izadi, Geraldine Fitzpatrick, Yvonne Rogers, Tom Rodden, The introduction of a shared interactive surface into a communal space, *Proceedings of the 2004 ACM conference on Computer supported cooperative work*, pp.49-58, 2004.