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

# 仮想ワークスペース上でデスクトップ画面の 共有・統合を行う共同作業支援システム

## 金衍廷

(コンピュータサイエンス専攻)

指導教員 田中二郎

2009年 3月

#### 概要

コンピュータの普及により、我々はコンピュータに囲まれた環境で生活し、コンピュー タを用いて作業を行うようになった。大抵のコンピュータは、ユーザに情報を提示するた めにディスプレイ装置を備えている。さらに、ディスプレイの低価格化により、一台のコ ンピュータに複数のディスプレイを用いることも一般的になっている。また、ミーティン グルームや教室などの場所には大画面が置かれ、街中には電子広告も増えている。このよ うなディスプレイのあふれる環境において、複数人のユーザが共同作業を行うとき、ある デスクトップ画面を任意のディスプレイに表示することは有意義である。

本研究では、仮想的な平面上にグループメンバのデスクトップ画面を配置し、その中で デスクトップ画面同士を重ねたり、結合したりすることができるインタフェースを提案し、 プロトタイプである VIWORD の実装を行った。ユーザはコンピュータを意識せずに、デ スクトップ画面を対象とした直感的な操作で、他のユーザの作業状況の確認、デスクトッ プ画面の転送、任意のデスクトップ画面の遠隔操作、ファイルの共有を行うことができる。

# 目次

第1章	序論
1.1	背景と問題点1
1.2	研究の目的
1.3	論文の構成
<i>ba</i> . 1.	
第2章	システムの設計6
2.1	システムの利用シーン
2.2	仮想ワークスペースインタフェース
2.3	仮想ワークスペースの操作8
2.3	5.1 デスクトップ画面の転送8
2.3	9.2 デスクトップ画面の結合
2.3	.3 ファイルの転送
第3章	ブロトタイプの実装: VIWORD
3.1	実装方針
3.2	システムの構成
3.3	仮想ワークスペースの表示
3.3	5.1 仮想ワークスペース上のコンポーネント
3.3	9.2 ズームと視点移動
3.3	3.3 表示の切り替え
3.4	仮想ワークスペースの操作
3.4	1 ピアパネルの操作
3.4	2 レプリカパネルの操作
3.4	3 ファイルパネルの操作
3.5	新規ユーザの参加
3.6	デスクトップ画面の転送
3.7	入力のリダイレクト
3.8	デスクトップ画面の結合
3.9	ファイルの転送
the a star	
弗4早 47	VIWOKD の利用ンテリオ ····································
4.1	一人 じの利用例
4.2	多人数での利用例

4.3	共用画面を含む利用例
4.4	複数の画面を結合して大画面を構成する利用例
第5章	関連研究
第6章	議論
6.1	本研究の位置付け
6.2	本研究の貢献
6.3	今後の展望
第7章	結論
謝辞	
参考文献	

# 図目次

义	1-1	仮想デスクトップの例 (Spaces・MSVDM)
义	1-2	商用のプロジェクタによる複数コンピュータの表示
义	1-3	マルチディスプレイの例
义	1-4	想定の利用シーン
凶	2-1	仮想ワークスペースの構想概念図 ·······7
凶	2-2	テスクトップ画面の転送
凶	2-3	デスクトップ画面の結合 ····································
义	2-4	結合されたデスクトップ画面へのデスクトップ画面の転送10
义	2-5	ファイルの転送
V	२-1	VIWORD におけるコンピュータ問の接続方式19
N	3-9	システムの構成
N N	3-3	ビアパネル
区 図	2-4	レプリカパネル
区 図	9-5	レノノルパマル 15
	00 9-6	アノアルハル $13$ ピアパラルの役動(前・後)
	00 9-7	(17) (前一後) 17 ピアパラルの社会(前・後) 17
区 図	ວ / ງ.ວ	レプリカウィンドウ···································
	ათ ე_0	レプリカウオントウ $18$
凶 団	3-9 9-10	レノリルハイルの主成(前・後) 19 レプリカパネルの投動(前・後)
凶	3-10 9-11	レノリルハイルの移動(削・仮) 19
区 図	3-11 9 1 9	レノリカハネルの有云(則・仮)
凶 団	3-12 0.10	レノリカハネルのノルリイスモート (削・仮)
区	3-13	ノアイルハイルの生成(削・仮)
× ×	3-14	URL ハイルの生成(則・仮)
凶	3-15	ファイルハイルの移動(削・後)
凶	3-16	ファイルバネルの転送(前・後)
凶	3-17	ファイルパネルの消去(前・後)
凶	3-18	新規ユーザの参加処理の流れ
义	3-19	デスクトップ画面の転送処理の流れ
义	3-20	デスクトップ画面の結合による遠隔操作の開始
汊	4-1	VIWORD 利用例 (一人のユーザ)

図 4-2	VIWORD 利用例	(多人数のユーザ)	2
🗵 4-3	VIWORD 利用例	(共用画面あり)3	<b>5</b>
図 4-4	VIWORD 利用例	(結合による大画面構成)3	7

## 第1章 序論

### 1.1 背景と問題点

ミーティングルームや教室などにプラズマディスプレイやプロジェクタが設置されてい ることが多い。本論文では、それらのディスプレイのことを共用画面(Shared display) と呼ぶことにする。共用画面は一台のコンピュータとディスプレイケーブルを通して接続 させ、その一台のコンピュータのデスクトップ画面をそのまま表示することができる。ミ ーティングの参加者は全員で同じものに注目でき、自然に意識統一が図られるため、ミー ティングの能率が向上する効果がある。

しかし、一般的なミーティングルームには共用画面が一つしか置かれてなく、他のコン ピュータのデスクトップ画面を共用画面に表示するためにディスプレイケーブルを差し替 える必要がある。さらに、議題がまったく別のものに変わるのではなく、関連する情報を 見せた後に元の画面に戻るなら、もう一度ディスプレイケーブルを差し替えなければなら ない。

共用画面に表示するソースコンピュータの切り替えを簡単にするならば、商用のディス プレイ切替器を用いてもある程度は解消できる。しかし、この方法は共用画面に表示した いコンピュータが予め決まっていることを前提として有用である。通常、共用画面に表示 するコンピュータを切り替えたい要望は突発的に起こるため、その場のすべてのコンピュ ータをディスプレイ切替器に接続させておかない限り、想定していなかったコンピュータ のデスクトップ画面を表示するためにはディスプレイケーブルの切り替えを避けられない。 共用画面をミーティングの参加者の数だけ用意すれは、ソースコンピュータの切り替えを 行わず複数台のコンピュータのデスクトップ画面を同時に見ることができるが、現実的で はない。

よりスマートな解決法として、ソフトウェアによりソースコンピュータを切り替えるこ とが考えられる。幸い、ネットワークを備えたミーティングルームは珍しくない。この場 合、共用画面にはソースコンピュータの切り替えを行うための専用のコンピュータが必要 となる。ミーティングの参加者は各自のコンピュータを共用画面に接続させておき、必要 に応じて共用画面に表示するコンピュータを選択すれば良い。

この方法のもう一つの利点として、共用画面に複数のデスクトップ画面を同時に表示さ せることが考えられる。例えば、二つのデスクトップ画面を比較して見たい場合に、表示 するコンピュータを繰り返して切り替えるなら、参加者の集中度を低下させてうまく比較 できない。しかし、同時に複数台のコンピュータのデスクトップ画面を比較して見ること ができれば、ミーティングが途切れることなくスムーズに対応できる。 共用画面に複数のデスクトップ画面を表示させることは**仮想デスクトップ**によく似ている。 仮想デスクトップとは、一台のコンピュータに接続された一つのディスプレイに、独立し た複数の仮想的なデスクトップ画面を切り替えて表示できるシステムを指す。GNOME や KDE などの X Window System 環境ではデフォルトで利用でき、Mac OS X では Spaces<sup>†1</sup> という機能で提供している。Windows XP では Virtual Desktop Manager for Windows XP (MSVDM)<sup>†2</sup> というツールとして提供している。(図 1-1)



図 1-1 仮想デスクトップの例 (Spaces・MSVDM)

共用画面に複数台のコンピュータのデスクトップ画面を表示することが、仮想デスクト ップと決定的に異なる点は、共用画面に表示されるデスクトップ画面がそれぞれ別のコン ピュータのデスクトップ画面であることである。それぞれのコンピュータはデスクトップ 画面をリアルタイムに共用画面へ送信することで実現できる。

これは VNC [15] のようなリモートデスクトップシステムによく似ている。VNC はク ライアント (デスクトップ画面を表示する側) がサーバ (デスクトップ画面を転送する側) ヘデスクトップ画面の転送を要求する。しかし、共用画面に複数台のコンピュータのデス クトップ画面を表示させる場合は、ソースコンピュータが共用画面に対して表示を要求す るので、方向性が正反対であるといえる。

単に、共用画面に複数台のコンピュータのデスクトップ画面を表示するだけなら、商用 のプロジェクタ<sup>†3</sup>を用いても可能である(図 1-2<sup>†4</sup>)。しかし、共用画面に表示するデス クトップ画面の数に制約があり、レイアウトを自由に変更することができず、より柔軟な 使い方ができない。

<sup>&</sup>lt;sup>†1</sup> http://www.apple.com/macosx/features/spaces.html, accessed January 2009

 $<sup>^{\</sup>dagger 2}$  http://www.microsoft.com/windowsxp/downloads/powertoys/xppowertoys.mspx, accessed January 2009

<sup>&</sup>lt;sup>†3</sup> 例えば、松下電器産業(株) TH-LB10NT

 $<sup>^{\</sup>dagger 4}$  http://plusd.itmedia.co.jp/pcupdate/articles/0403/29/news047.html, accessed January 2009



図 1-2 商用のプロジェクタによる複数コンピュータの表示

ミーティングの場合を含めてより一般的な状況を考えてみると、単にデスクトップ画面 を集めて表示するだけでなく、他のコンピュータを遠隔操作したい要望がよくある。例え ば、ソフトウェアを開発するグループで、共用画面にソースコードを表示してコードレビ ューを行っている場合、言葉で説明するより直接書き直したい場合などが考えられる。

以上で、一つの共用画面に複数台のコンピュータのデスクトップ画面を表示することの 利点とその実現方法について述べたが、それとは逆に、複数のディスプレイに一台のコン ピュータのデスクトップ画面を表示することも考えられる。実は、これは日常的に見られ ることであり、マルチディスプレイがまさにそれに当たる。

マルチディスプレイ環境では、カーソルやウィンドウがディスプレイの境界を通り抜け て移動できる(図 1-3)。当然ながら、別のコンピュータに接続している二つのディスプレ イが隣り合っていても、通常そのようなことはできない。マルチディスプレイの隣に別の コンピュータに接続したディスプレイが置かれたとすると、マルチディスプレイに使い慣 れたユーザは独立しているディスプレイにカーソルを持って行けないことに一瞬戸惑って しまう。また、別のコンピュータに接続しているディスプレイが隣り合っていて、それぞ れのマウスが置かれている場合、ユーザはどのマウスがどのディスプレイに対応している かで悩んでしまうこともある。特に、同種のワイヤレスマウスならより困惑しやすい。

マルチディスプレイを利用するとき、各ディスプレイにデスクトップ画面をどのように 配置するかはとても重要である。もし、実世界によるディスプレイ間の位置関係と、設定 上のデスクトップ画面の位置関係がずれていたら、ユーザは強い困惑を感じる。最悪の場 合、横並びの二つのディスプレイが中心部の境界ではなく、両端の境界でつながるように 設定されていたとしたら、ユーザは強いストレスを感じる。



図 1-3 マルチディスプレイの例

## 1.2 研究の目的

本研究を始めたきっかけとして、一つの共用画面に複数台のコンピュータのデスクトップ画面を表示することでグループの共同作業(Collaboration)を支援することがある。その矢先に、どのディスプレイにどのコンピュータのデスクトップ画面を表示するかという問題と、デスクトップ画面の間の位置関係を柔軟に構成する方法について検討するようになった。

本研究の目的は、グループの共同作業を支援するための、各ユーザのデスクトップ画面 の間の包含関係と結合関係をダイナミックに再構成可能なシステムを開発することである。 共同作業の中でも、主にミーティングルームや教室のような、共用画面の設置された場所 における共同作業を想定している(図 1-4)。また、共用画面がない場所で偶然に集まった ユーザ同士での、予期せぬ共同作業も考慮したシステム設計を行う。そして、開発したシ ステムが共同作業を支援するにあたって有用であると思われる利用シナリオをいくつか挙 げて説明する。



図 1-4 想定の利用シーン

## 1.3 論文の構成

第2章では本研究の目的を達成するためのシステムの設計について述べ、第3章ではプ ロトタイプの実装について説明する。第4章ではシステムの利用シナリオをいくつか挙げ て有用性の検討を行い、第5章で関連研究について述べる。第6章で本研究の位置付けや 貢献、今後の展望について議論し、第7章で本研究の結論を述べる。

## 第2章 システムの設計

### 2.1 システムの利用シーン

共同作業を行うグループはいつも固定の場所に留まることはない。研究室やオフィスに など、主な作業を行う場所が決まっている場合もあるが、それでもミーティングの際には、 別途のミーティングルームで行うことが多い。さらに、授業やコンファレンスのように、 たまにしか集まらない場合や、偶然集まった場所で共同作業を行う場合も考えられる。こ のように、共同作業が行われる場所や参加するユーザが変動する中で、可能な限り多くの ケースを考慮し、いつもどこでも素早く共同作業を開始できることが望ましい。

以上のことを踏まえ、共同作業に参加するユーザのコンピュータ同士を P2P (Peer to Peer) 方式で相互接続させることにする。ユーザはアプリケーションを起動するだけで、 即座に共同作業に参加できるようにする。サーバを要しないので場所は選ばないが、ネットワークが利用できることを前提とする。

さらに、研究室やオフィスなどの主な作業場所に、外部の人が入り込むことはないはず であり、ネットワーク自体にセキュリティの設定がなされることが多いので、ユーザがア プリケーションを起動する際にはログインを要求しないことにする。これには、共同作業 への素早く参加できることと、ユーザの負担を軽減できるといったメリットがある。

## 2.2 仮想ワークスペースインタフェース

共同作業に参加しているユーザであっても、通常はいつも通りの個人作業に専念し、時 折、他のユーザの作業机に足を運んで作業状況を確認するのが普通である。研究室やオフ ィスなどの場においてはコミュニケーションのきっかけとなり、互いの進捗状況を把握で きるといった肯定的な効果がある。一方で、他の人の作業状況は見たいが、作業の妨げに はなりたいために、会話に発展たくない場合もある。

以上のことを踏まえ、いつでも他のユーザの作業状況を確認できる**仮想ワークスペース** (Virtual Workspace)を提供する。仮想ワークスペースには共同作業に参加しているすべて のユーザのデスクトップ画面が配置され、ユーザは仮想ワークスペースの中で、他のユー ザのデスクトップ画面を見ることができる(図 2-1)。

ユーザはアプリケーションを起動すると同時に、自分のデスクトップ画面をグループメ ンバ全員に公開することになる。共同作業を行うコンピュータは個人で使用するものでは なく、あくまでも作業専用でことを想定する。したがって、親しみのあるグループでの共 同作業では、デスクトップ画面を見られることに対する抵抗感や、プライバシーの侵害は ないとみなす。

仮想ワークスペースを均等に分割して各ユーザのデスクトップ画面を配置するといった 選択肢もある。しかし、参加ユーザの数が n<sup>2</sup> (n は自然数)以外の場合、画面に余分のス ペースを残さないためにはデスクトップ画面のアスペクト比を維持できなくなり、逆に、 アスペクト比を維持するためには余分のスペースが残るといった問題がある。したがって、 各ユーザのデスクトップ画面を、サイズを維持したまま、仮想ワークスペース中のどこに でも配置できるようにする。



図 2-1 仮想ワークスペースの構想概念図

仮想ワークスペースはズーム機能を持つべきである。デフォルトのズームレベルはすべ ての参加ユーザのデスクトップ画面が見えるようにする。そのため、各デスクトップ画面 は元のサイズより縮小して表示され、どのようなアプリケーションウィンドウを起動して いるか程度の作業状況しか得られない。多くの場合、それくらいの情報量で十分かも知れ ないが、より詳細に見たい場合は、ズームレベルを上げられるようにする。

各ユーザのデスクトップ画面の配置位置は、どのユーザにでも変更できるものとする。 あるユーザが任意のデスクトップ画面の位置が変更した場合、すべてのユーザの仮想ワー クスペースにそれが反映される。ユーザ間で仮想ワークスペースの同期を取る理由は、仮 想ワークスペース内のデスクトップ画面の配置そのものが、意味を持つことがあるからで ある。ユーザ達は実世界のディスプレイの配置と仮想ワークスペースのデスクトップ画面 の配置を自然に一致させるだろうという強い確信を持っている。

### 2.3 仮想ワークスペースの操作

#### 2.3.1 デスクトップ画面の転送

ユーザはコンピュータの操作に困ったとき、他のユーザに自分のコンピュータを操作す ることを依頼する場合がある。例えば、あるアプリケーションの操作の仕方がわからない 場合や、コンパイルエラーの原因がわからない場合などがすぐ想像できる。このような場 合に、通常のオフィスや研究室では、他のユーザを自分の作業机のところへ呼ぶか、ノー ト型 PC ならそれを他のユーザのところへ持って、操作してもらうしかない。

システムによってサポートされている環境では、仮想ワークスペースの中でソースコン ピュータ(画面転送を行う側)のデスクトップ画面をターゲットコンピュータ(画面転送 を受ける側)のデスクトップ画面へドラッグ&ドロップするだけの簡単な操作により、あ るデスクトップ画面を任意のデスクトップ上に表示させることを許容する(図 2-2)。そし て、そのウィンドウの中でキーボードとマウスを用いてソースコンピュータを遠隔操作す ることが可能である。



図 2-2 デスクトップ画面の転送

さらに、仮想ワークスペースにはデスクトップ画面の転送の状況がわかるような表示が される。ターゲットデスクトップ画面上に転送されたデスクトップ画面の位置とサイズに 相当する矩形と、その矩形とソースデスクトップ画面を結ぶ矢印を表示する。そして、矩 形をドラッグしてターゲットデスクトップ画面の中での位置を変更することができ、もし ターゲットデスクトップ画面の外側にドラッグされたら、デスクトップ画面の転送を終了 させる。

以上のすべての操作は、ソースコンピュータのみでなく、共同作業に参加しているユー ザなら誰でもできる。これは、共用画面があった場合を考慮したもので、共用画面にどの デスクトップ画面を表示させるかを誰もが自由にコントロールできるようにするためであ る。

#### 2.3.2 デスクトップ画面の結合

仮想ワークスペース上でデスクトップ画面同士を隣り合うように配置することで、デス クトップ画面同士を結合することが考えられる(図 2-3)。一台のコンピュータで複数のデ ィスプレイを利用するマルチディスプレイによく似ている機能である。それぞれのデスク トップに繋がっているコンピュータが同一のものであるか、異なるものであるかの違いで あるが、ユーザにとって重要なのは作業を行うデスクトップであり、コンピュータ本体で はない。ユーザにコンピュータ本体の存在を意識させず、自由にデスクトップを利用して もらうことが大事なのである。



図 2-3 デスクトップ画面の結合

仮想ワークスペース上には、結合されているデスクトップ画面を囲む矩形が表示される。 ユーザは、結合されたデスクトップ画面間で物理的な境界を越えカーソルを持っていくこ とができ、そこでマウスとキーボードの操作を行うことができる。結合を解除するために は、デスクトップ画面を他のデスクトップ画面に接しないところへ移動させれば良い。

デスクトップ画面の結合する機能は、複数台のコンピュータを持つユーザが、特によく 利用すると考えられるが、もうひとつ重要な使い方がある。それは、複数台の共用画面を 結合させて大画面(Large display)を構成することである。一般に、大画面を一つのデバ イスで構成することはコストが高く、複数のデスクプレイを組み合わせて一つの大画面を 構成することがよくある。デスクトップ画面を結合させると、一つのデスクトップ画面の ように扱うことができる。すなわち、結合されたデスクトップ画面に他のデスクトップ画 面を転送することができるのである(図 2-4)。さらに、転送されたデスクトップ画面のサ イズを結合されたデスクトップ画面のトータルサイズに合わせて拡大すれば、共用大画面 が完成する。



図 2-4 結合されたデスクトップ画面へのデスクトップ画面の転送

#### 2.3.3 ファイルの転送

共同作業を行う際に、ファイルをグループメンバ同士で共有したいことがよくある。例 えば、ミーティングを行う際にグループメンバ全員に資料を配りたいとか、ソフトウェア を開発するグループで作成した各種のドキュメントや UML データを共有したいとかいう 要望がよくある。通常はファイルサーバにアップロードしておくか、USBメモリなどの外部記憶装置を用いる場合が多い。

ユーザは自分のデスクトップからファイルアイコンをドラッグ&ドロップして仮想ワー クスペース上にファイルを置くことができる。このとき、仮想ワークスペースにはどのユ ーザが置いたファイルであるかわかるように矢印が表示される。そして、他のユーザにそ のファイルを転送するためには、仮想ワークスペース上でファイルを転送したいユーザの デスクトップ画面へドラッグ&ドロップすれば良い。いったん仮想ワークスペース上に置 いたファイルは、ファイルの持ち主ユーザや転送先のユーザでなくても転送を行うことが できる。ファイルの転送が行われると、どのユーザのデスクトップに転送したかわかるよ うに、仮想ワークスペース上にはファイルパネルと転送先のデスクトップ画面を結ぶ矢印 が表示される(図 2-5)。ファイルの位置はドラッグ&ドロップで移動できるが、それ以上 転送することがなくなった場合は、持ち主ユーザのデスクトップ画面の中へドラッグ&ド ロップすれば、仮想ワークスペースから削除される。



図 2-5 ファイルの転送

## 第3章 プロトタイプの実装:VIWORD

本章では、第 2 章で述べた設計に基づいて実装したプロトタイプ **VIWORD** (<u>VI</u>rtual <u>Workspace ORganizing Desktops</u>) について説明する。

## 3.1 実装方針

グループの共同作業は様々な場面がある。決まった場所で決まったユーザ同士で行うこともあり、予期せぬ場所で偶然出会ったユーザ同士で共同作業を始めることもある。様々な状況においても共同作業を即時開始できるようにするため、VIWORD はサーバを置かずに P2P 方式で相互接続させる(図 3-1)。この方法では、デスクトップ画面の転送方式によって加減はあるものの、参加するコンピュータの数が増えるにつれて通信量も増えるので、拡張性が低いといえる。ただし、100Mbpsの帯域幅のネットワークを用いる場合、10人以下の小規模なグループでの利用には持ちこたえられる。



図 3-1 VIWORD におけるコンピュータ間の接続方式

共同作業に用いるコンピュータがすべて同じ OS で稼働するとは限らない。様々な OS に対応するため、VIWORD は Java 言語を用いて実装を行った。Java Standard Edition Runtime Environment 1.6.0 以上で起動可能である。また、実装を簡単に行うために、コ ンピュータ間の通信には RMI (Remote Method Invocation) を用いている。 VIWORD はマニュアルの要らないシステムを目指し、インタフェースはシンプルかつ 直感的なものにする。仮想ワークスペースをグラフィカルに表示する**仮想ワークスペースビ ユー**だけを提供し、メニュなどを用いた煩雑な設定などは一切排除する。ユーザは、仮想 ワークスペースビューの中に配置される各種コンポーネントをマウスで操作するだけで、 VIWORD のすべての機能を駆使することができる。

### 3.2 システムの構成

VIWORD の主要な処理は、仮想ワークスペースビューにおけるユーザの操作を、他の ユーザ達の仮想ワークスペースビューに反映することである。そこで必要となるのは、**仮** 想ワークスペース表示部とネットワーク部である。さらに、デスクトップ画面の転送やマウ スやキーボード入力のリダイレクトを行うための画面転送・遠隔操作部がある(図 3・2)。 ネットワーク部には、VIWORD 間のほとんどの通信を行うための、リモートオブジェク トである Peer クラスと、そのリモートオブジェクトがあるコンピュータを見つけ出すた めの UDP 通信を行う Connector クラスがある。各構成要素の詳細については次のセクシ ョンから説明する。



図 3-2 システムの構成

## 3.3 仮想ワークスペースの表示

VIWORD は仮想ワークスペースを表示する仮想ワークスペースビューをユーザ提供する。本来、仮想ワークスペースは無限の広さを持つべきであるが、プロトタイプの実装においては簡略化のために、8192×6144 に固定にしている。これは、現在最も一般的なノート型 PC のディスプレイサイズである 1024×768 の縦横 8 倍であるが、10 人以下の少人数グループで利用する分には十分な広さである。

#### 3.3.1 仮想ワークスペース上のコンポーネント

仮想ワークスペース上には、数種類のコンポーネントが共同作業の状況に応じて表示される。まずは、各ユーザのデスクトップ画面をリアルタイムで表示するピアパネルがある (図 3-3、VIWORD の仮想ワークスペースビューの背景色は黒であるが、本章では見や すくするため灰色にしてスナップショットを撮ったものを掲載する)。ピアパネルは共同作 業に参加しているユーザの数だけ存在し、ピアパネル同士で重なり合うことのないように 配置される。また、ピアパネルの右上には、どのデスクトップ画面に対応しているかを表 すために、「<ユーザ名> @ <コンピュータ名>」が表示される。



図 3-3 ピアパネル

次に、あるユーザのデスクトップ画面を他のユーザのデスクトップ上に転送しているこ とを表す、半透明のレプリカパネルがある(図 3-4)。レプリカパネルはデスクトップ画面 の転送先ピアパネルの上に重なって表示される。また、レプリカパネルと転送元のピアパ ネルは常に矢印で結びつけられる。



図 3-4 レプリカパネル

最後に、ファイルを仮想ワークスペースビューへドラッグ&ドロップすると表示される、 ファイルパネルがある(図 3-5)。ファイルパネルの内側に表示される画像はファイルに関 連付けられたアイコンを利用するため、ユーザ同士で異なることがある。ファイルパネル は常に転送元のピアパネルと矢印で結びつけられ、他のユーザ達へファイルの転送を行っ た場合は、転送先のピアパネルとも矢印で結びつけられる。また、ファイルパネルの右上 にはファイル名が表示される。



図 3-5 ファイルパネル

#### 3.3.2 ズームと視点移動

共同作業を行うグループのすべてのユーザのデスクトップ画面を表示するために、仮想 ワークスペースを縮小表示している。ユーザが仮想ワークスペースビューを開くと、すべ てのピアパネルが見られる程度にズームレベルが自動設定される。そのままでも各ユーザ がどのようなアプリケーションをアクティブにしているか程度の作業状況は把握できるが、 作成しているドキュメントの内容が読める程度の、より詳細なところまでを確認するため には、ズーム機能が不可欠である。ズームはズームボタンやスライダーを用いることも考 えられるが、簡単かつシンプルなインタフェースを実現するために、VIWORD はマウス ホイールによる仮想ワークスペースビューのズーム機能を提供する。また、設定可能なズ ームの倍率は 1/8(デフォルト)、1/4、1/2、1/1 の 4 段階ある。

ユーザがズームレベルを上げると、仮想ワークスペースがディスプレイに納まらなくな り、視点移動が必要となる。視点移動はスクロールバーを用いることも考えられるが、簡 単かつシンプルなインタフェースを実現するために、VIWORD はマウスドラッグによる 仮想ワークスペースビューの視点移動機能を提供する。仮想ワークスペース上のコンポー ネントに対する操作と混乱しないように、視点移動は右ボタンで行う。

マウスホイールによるズームと右ボタンのドラッグ&ドロップによる視点移動は、直感的な操作を可能にするが、現在ディスプレイに見える領域が全体の仮想ワークスペースの中で、どの部分に相当するかが分からないといった問題がある。この問題を解決し操作性を向上させるために、VIWORD は仮想ワークスペースのミニマップを提供する。ミニマップは、ズームレベルや視点移動に影響されず、常に仮想ワークスペースビューの左上に位置し、ディスプレイに表示されている領域が仮想ワークスペース全体のどの部分であるかを表示する (図 3.6)。さらに、ミニマップから視点移動を行うことも可能である。

#### 3.3.3 表示の切り替え

共同作業を行う最中にも、ユーザは個人作業にほとんどの時間を費やす。他のユーザの 作業状況を確認したり、デスクトップ画面を転送したりすることはたまにしか行われない。 VIWORD が起動したとき、仮想ワークスペースビューはデフォルトでアイコン化された 状態である。ユーザは必要に応じて仮想ワークスペースビューを可視化することができる。

ユーザが仮想ワークスペースビューを可視化するとき、デスクトップ画面がピアパネル に変化するようなアニメーション表示を行う。これは、自分のディスプレイが仮想ワーク スペースの一部であることを、ユーザに強く意識させるためである。

#### 3.4 仮想ワークスペースの操作

#### 3.4.1 ピアパネルの操作

新規ユーザが共同作業に参加したとき、そのユーザのピアパネルが配置される位置は、 他のピアパネルと重ならないようにしてランダムで決定される。ユーザ達は、実世界にお けるディスプレイの位置関係と直感的に対応づけられるように、仮想ワークスペースの中 のピアパネルを再配置することが期待される。そのために、VIWORD は仮想ワークスペ ース内でピアパネルを移動することを許容する。

ピアパネルを移動させるには、移動させたいピアパネルの上で左ボタンを押しながらド

ラッグし、配置したい位置でドロップすれば良い。ドラッグ中に移動の処理を行うと、他のコンピュータとの通信も行われてスムーズなドラッグができないため、ドロップしたときのみピアパネルの移動を行う必要がある。ドラッグ中のピアパネルの周囲にハローが表示され、ドロップしたらどの位置に移動されるかを表す、半透明のゴーストパネルが現れる。 そして、ゴーストパネルとその元となるピアパネルの間には矢印が描かれる(図 3-6)。



図 3-6 ピアパネルの移動(前・後)



図 3-7 ピアパネルの結合(前・後)

ー台のコンピュータに複数台のディスプレイを接続させてデスクトップを拡張するマル チディスプレイのように、それぞれ異なるコンピュータのデスクトップを自由にカーソル 移動して操作することを、VIWORD はあるトリックを用いて実現する。仮想ワークスペ ースビューの中でピアパネルを結合させて通常の画面に戻ると、まるでマルチディスプレ イのようにディスプレイの境界を越えてカーソルを移動させることができる。

ピアパネルを結合させるには、他のピアパネルと上下左右の内どれかで隣り合うような

位置へ移動させれば良い。しかし、ピアパネル同士をぴったり隣り合わせることは、ユー ザにとって至難の技である。それを手助けするために、ピアパネルのドラッグ中にピアパ ネルの結合が起こるような位置とある距離 <sup>†5</sup> 以内に入ったら、ゴーストパネルを強制的 にその位置へ移動させている。ピアパネルが結合される位置にゴーストパネルが位置した とき、それのことをユーザに伝えるために、ゴーストパネルの色を変更させ、結合される ピアパネルすべてにかけてハローを表示する。そして、ユーザがドロップを行ってピアパ ネル同士が結合されたら、そのことを伝えるために色違いのハローが表示される(図 3-7)。 共同作業において、あるユーザのデスクトップ画面を他のユーザのデスクトップ上に転 送し、遠隔操作を行えるようにすることは要望が高い。この一般的な要望を満たすため、

VIWORD では仮想ワークスペース上にレプリカパネルを生成し、そのレプリカパネルと 重なるピアパネルの、元のデスクトップ上にレプリカウィンドウを生成することで、デスクト ップ画面を表示させることを許容する(図 3-8、図 3-9 と比較して参照)。レプリカウィ ンドウに描画されるデスクトップ画面はリアルタイムで更新され、その中で発生するマウ スとキーボードの入力イベントは、画面転送元のコンピュータへリダイレクトされる。



図 3-8 レプリカウィンドウ

レプリカパネルを生成するには、画面転送元となるピアパネルをドラッグして画面転送 先となるピアパネルと重ねれば良い。ピアパネルをドラッグしてゴーストパネルが他のピ アパネルと重なると、そのことをユーザに伝えるためにゴーストパネルの色を変更させる。 そこでユーザがドロップを行うと、ゴーストパネルのあった位置にレプリカパネルが生成 される(図 3-9)。レプリカウィンドウと自分のデスクトップとの境界には、赤色のボーダ ーラインが表示される。

<sup>\*5</sup> 現在の実装でピアパネル同士での引力の働く距離は、ズームレベルに関係なく 20 ピクセル



図 3-9 レプリカパネルの生成(前・後)

### 3.4.2 レプリカパネルの操作

レプリカウィンドウを移動させたい場合は、対応するレプリカパネルをドラッグ&ドロ ップで移動させれば良い。レプリカパネルは特定のピアパネルに従属しないため、同じピ アパネルの中でしか移動できない訳ではない。すなわち、他のピアパネルに移しても良く、 さらに、同時に複数のピアパネルに重なっても良い(図 3-10)。もし、ピアパネルの結合 によって複数のディスプレイから構成された共用画面があって、その結合されたピアパネ ルに対してレプリカパネルを配置する場合でも、一つのデスクトップ画面の場合と同等な 扱いができる。レプリカパネルの移動に対する唯一の制約は、少なくとも一つのピアパネ ルと重ならなければならないことだけである。



図 3-10 レプリカパネルの移動(前・後)

デスクトップ画面の転送をやめてレプリカパネルを消去したい場合は、レプリカパネル

の移動に対する制約を満たさないようすれば良い。すなわち、どのピアパネルとも重ならない位置にドラッグ&ドロップすることである(図 3-11)。

レプリカウィンドウを転送先のデスクトップで全体表示にするために、レプリカパネル をピアパネルとぴったり重ねることは至難の技である。さらに、転送先のデスクトップサ イズが転送元と異なる場合は不可能なことである。実際、結合されたデスクトップ上にレ プリカウィンドウを全体表示させたい要望はよくある。VIWORD はこの問題を解決する ために、レプリカパネルをフルサイズモードにすることを許容する。レプリカパネルをフル サイズモードにするためには、レプリカパネルをダブルクリックすれば良い。フルサイズ モードにすると、レプリカパネルと重なっているすべてのピアパネルとその結合されたピ アパネルを覆うように位置とサイズが変更され、それに対応するレプリカウィンドウにも 適応される(図 3-12)。また、もう一回ダブルクリックすると、通常のサイズに戻る。



図 3-11 レプリカパネルの消去(前・後)



図 3-12 レプリカパネルのフルサイズモード(前・後)

#### 3.4.3 ファイルパネルの操作

VIWORD は、仮想ワークスペース上にファイルをアップロードして、他のユーザ達へ 転送することを許容する。ユーザが通常のデスクトップ画面から共有したいファイルをド ラッグして、仮想ワークスペース上でドロップすると、ドロップした位置にファイルパネ ルが生成される(図 3-13)。VIWORD のすべてのコンポーネントに対する操作の中で、 このファイルパネルの生成だけは特別で、ファイルの持ち主ユーザ以外のユーザは行うこ とができない。また、仮想ワークスペース上にファイルパネルが置かれただけでは、まだ ファイルのデータの転送は行われず、ファイル名を共同作業のグループに公開しているだ けである。



図 3-13 ファイルパネルの生成(前・後)

ファイルパネルの拡張した URL パネルもある。URL パネルは文字列が仮想ワークスペ ースビュー上にドラッグされ、その文字列が URL と判断された場合に生成される(図 3-14)。他のユーザ達に見てもらいたいウェブページを見つけた場合、アドレスやアドレス バーの左に表示されるアイコンをドラッグして、仮想ワークスペース上に配置できる。 仮想ワークスペース上のファイルパネルの移動は、ピアパネルと同じようにして行われ る(図 3-15)。すなわち、ドラッグを開始するとゴーストパネルが現れ、すべてのピアパ ネルと重ならない位置でドロップするだけである。URL パネルの移動も同様である。



図 3-14 URL パネルの生成(前・後)



図 3-15 ファイルパネルの移動(前・後)

ファイルパネルを他のユーザのピアパネルの上に重なるようにしてドラッグすると、ゴ ーストパネルの色が変わり、ファイルの転送が可能であることを表す。そのままドロップ すると、実際にファイルデータの転送が行われ、ファイルパネルと転送先ピアパネルの間 に矢印が描かれる(図 3-16)。これにより、どのユーザからどのユーザへファイルの転送 が行われたかといった履歴までをグループのユーザ間で共有できる。ファイルの転送が終 了すると、転送先の VIWORD は転送されたファイルのあるフォルダを開く。また、URL パネルの転送が終了すると、その URL のウェブページがブラウザで開く。



図 3-16 ファイルパネルの転送(前・後)

仮想ワークスペースからファイルパネルを消去したい場合は、ファイルパネルを生成したユーザのピアパネルと重なるようにドラッグすれば良い。このとき、ファイルの転送履歴も同時に消去される(図 3-17)。



図 3-17 ファイルパネルの消去(前・後)

### 3.5 新規ユーザの参加

新規のユーザが、共同作業のオーバレイネットワークに参加するためには、少なくとも ー台のコンピュータのネットワークアドレス(IPアドレス+ポート番号)を知らなければ ならない。一台のコンピュータのネットワークアドレスが分かれば、オーバレイネットワ ークに参加しているすべてのコンピュータのネットワークアドレスを取得できるからであ る。VIWORD で共同作業を行うためのオーバレイネットワークを形成するために、 VIWORD を稼働しているユーザから口頭でネットワークアドレスを聞き、そのアドレス を指定して VIWORD を起動する方法が考えられる。しかし、共同作業に新規参加するユ ーザが増えるたびにネットワークアドレスを教えなければならないということは、短時間 で簡単に共同作業を開始するといった実装方針に反するものである。したがって、ユーザ は VIWORD を起動するだけで、VIWORD が他の VIWORD を自動で探索してオーバレイ ネットワークに参加することが望ましい。VIWORD の新規ユーザの参加処理の流れを図 3-18 に示す。





VIWORD が起動して最初に行うのは、「初めまして」のメッセージ <sup>+6</sup> をブロードキャ ストすることである <sup>+7</sup> 。稼働している VIWORD が存在する場合、「よろしく」のメッセ ージ <sup>+8</sup> を返す。VIWORD の稼働しているコンピュータが複数台あった場合は、複数の返 事が届いても最初に到着した返事以外は破棄する。次に、返事をしてきたアドレスのコン ピュータから VIWORD の RMI リモートオブジェクトを取得する <sup>+9</sup> 。

そして、取得した既存の VIWORD のリモートオブジェクトに対して共同作業への参加 を要請すると、既存の VIWORD は新規の VIWORD の仮想ワークスペースを自分の仮想 ワークスペースと同等なものになるような処理を行う。まずは、すべての VIWORD のリ モートオブジェクトを伝え、ピアパネルを追加させる。次に、レプリカパネルとファイル パネルを追加させ、ファイルパネルに転送先がある場合はそのことも伝える。

最後に、新規参加の VIWORD は、自身のピアパネルを仮想ワークスペースに追加し、 他のすべての VIWORD にも追加させる。ピアパネルの最初の配置位置はこのとき決定さ れる。

### 3.6 デスクトップ画面の転送

デスクトップ画面を他の VIWORD へ転送する最も単純な方法は、一定時間ごとにスク リーンキャプチャを行って得た画像を、そのまま転送することである。実際に、現在の実 装では java.awt.Robot クラスを用いて 1 秒置きにスクリーンキャプチャを行っている。し かし、この単純な方法ではネットワークの負荷が大きく、共同作業に参加するユーザの数 が増えれば増えるほど、より深刻な問題となる。この問題を解消するためにいくつかの対 策を施している。VIWORD のデスクトップ画面の転送処理の流れを、図 3-19 に示す。

まずは、仮想ワークスペースビューは非表示になると、他の VIWORD が自分に対して デスクトップ画面を転送しないようにする。共同作業中にも、通常は仮想ワークスペース を開くことは少ないことから、これによる通信量の削減は大きい。

次に、ユーザごとに転送するデスクトップの画面のサイズを設定できるようにした。仮 想ワークスペースビューのズームレベルが変更されると、他のすべての VIWORD に変更 されたズームレベルを伝え、各 VIWORD は次回のデスクトップ画面の転送から、新しい ズームレベルに応じて画像を縮小して転送する。

また、デスクトップ画面を一つの画像ではなく、いくつかの領域に分けて転送するよう にした。ユーザがドキュメントの作成やコーディングなどの作業を行うときには、デスク トップ画面が1秒の短い間隔で、全領域に渡ってころころ変わることはほとんどないので、 この方法はある程度の効果を発揮する。ズームレベルに合わせて縮小されたデスクトップ の画像を、デスクトップ画面サイズの縦横1/8サイズの領域に分ける。すなわち、ズーム

<sup>&</sup>lt;sup>†6</sup> 現在の実装においては「viword\_hajimemashite」

<sup>†7</sup> 現在の実装におけるブロードキャスト用のポート番号は6789

<sup>&</sup>lt;sup>†8</sup> 現在の実装においては「viword\_yoroshiku」

<sup>†9</sup> 現在の実装における RMI レジストリ用のポート番号は 5678

レベルが1の場合は計64個の領域に分けられ、ズームレベルが1/8の場合は領域が1個のままとなる。

そして、前回の転送した同じ領域の画像と比較し、同じでないと判断された場合にのみ 次のステップへ進む。前回の画面転送と変わってない部分に対いては、転送するだけ無駄 であるからである。しかし、前後の2枚の画像をまともに比較すると、プロセッサの負荷 が増大し、他の作業を妨害することがある。現在の VIWORD の実装においては、すべて のピクセルの ARGB 値(integer 型)の排他的論理和を比較し、もし同じであれば、2枚 の画像が同じであると判断している。



図 3-19 デスクトップ画面の転送処理の流れ

最後に、画像転送の直前に javax.imageio.ImageIO クラスを用いて PNG <sup>†10</sup> 圧縮をか けるようにした。VIWORD が画像処理に用いている java.awt.image.BufferedImage オブ ジェクトは直列化(Object serialization)できないため、そのままでは転送することがで きなく、いったんピクセルデータ自体を byte 配列にする必要がある。PNG 圧縮はこのタ イミングで行われるが、この圧縮処理に若干の遅延が生じ、画像のサイズが大きければ大 きいほど遅延も長くなる。このことも、画像を分割する理由の一つになっている。

## 3.7 入力のリダイレクト

レプリカウィンドウ上でもマウスとキーボードの入力は、そのレプリカウィンドウの転 送元のコンピュータへリダイレクトされる。その仕組みは次のように単純なものである。

まず、レプリカウィンドウの中でマウスイベントとキーボードイベントが発生すると、 それを相手に伝える。次に、画面転送元の EventHandler クラスは java.awt.Robot クラス を用いて、マウスイベントやキーボードイベントを発生させる。

## 3.8 デスクトップ画面の結合

結合されたデスクトップ間ではカーソルの移動が可能になるが、実際にはそうであるか のように見せかけるだけである。このトリックのために、デスクトップ画面の周囲に幅1 ピクセルのフレーム(エッジフレーム)といった仕掛けが必要である。エッジフレームを 用いたデスクトップ間でカーソルを移動させるトリックは次のように行われる。

まずは、ピアパネルを移動させてデスクトップ画面の結合を行うと、結合されたユーザ のデスクトップ上には互いのレプリカウィンドウが表示される。ただし、そのレプリカウ ィンドウはディスプレイの外側に位置するために、デスクトップ画面の変化は見られない。 また、レプリカパネルにより表示されるレプリカウィンドウは元の画面サイズを保持する ことに対し、結合により表示されるレプリカウィンドウは転送先(表示される側)のデス クトップ画面のサイズと同じになるように拡大縮小される(図 3-20 上)。

次に、ユーザがカーソルをディスプレイの端(レプリカウィンドウの位置する側)に達 するように移動させると、対応するレプリカウィンドウがディスプレイ全体に表示される ように移動する。そして、カーソルは反対側の端に移動され、ユーザはカーソルが連続し て移動するように感じる。カーソルはレプリカウィンドウの上にあるので、デスクトップ 画面の転送元へ入力のリダイレクトが行われる(図 3-20下)。

当初は、結合されたデスクトップへの入力リダイレクトにレプリカウィンドウを用いる ことを考えていなかった。ダミーのウィンドウを全画面表示にして、自分のデスクトップ 画面をキャプチャした画像を張り付け、カーソルアイコンを透明な画像に変更することで、 自分のデスクトップからカーソルが消えたように見せかけることを検討していた。しかし、

<sup>&</sup>lt;sup>†10</sup> Portable Network Graphics

そうした場合は問題点が二つあり、一つは結合されたディスプレイが実世界においてユー ザの目に入らない位置にある場合がある点と、もう一つは入力リダイレクトが行われてい る事実をユーザにフィードバックできない点である。両方の問題を解決するために、 VIWORD はレプリカウィンドウを用いるような実装になっている。



図 3-20 デスクトップ画面の結合による遠隔操作の開始

## 3.9 ファイルの転送

ファイルを転送するために、まずファイル読み込み用のバッファ<sup>†11</sup>を用意する。そし て、バッファサイズずつファイルを読み込んで、オフセットとファイルから読み込んだデ ータのサイズと共に転送を行う。転送先の VIWORD はオフセットと送られたデータのサ イズを足してファイルサイズと同じになったらとファイル転送が終了したことがわかる。 転送されたファイルは規定のディレクトリ<sup>†12</sup> に格納される。もし、転送するファイルと 同じ名前のファイルが存在する場合は、既存のファイルはファイル転送の始めに削除され る。しかし、URL パネルの場合は、ファイル名の代わりに URL そのものを利用するため、 転送が開始されと同時に終了となる。

ファイルの転送が終了すると、ファイルパネルの場合はディレクトリを開き、URL パネルの場合はブラウザで URL のウェブページを開く。これは、両方とも java.awt.Desktop クラスの browse()メソッドを用いている。

<sup>†11</sup> 現在の実装におけるバッファサイズは 4096 byte

<sup>†12</sup> 現在の実装においてのファイル格納ディレクトリは

<sup>[</sup>java.lang.System.getProperty("user.home")+java.io.File.separator+"VIWORD"]

## 第4章 VIWORD の利用シナリオ

### 4.1 一人での利用例

パーソナルコンピュータの普及に伴い、一人のユーザが複数台のコンピュータを利用す ることは珍しいことではなくなっている。また、一台のコンピュータで複数のディスプレ イを利用するマルチディスプレイも一般的になっている。デスクトップ PC に二台のディ スプレイをつなげているユーザの場合、カーソルがディスプレイ間の境界を自由自在に行 き来できる。しかし、このユーザがノート型 PC を作業机の上に置いたとしても、二台の ディスプレイ間を行き来しているカーソルがノート型 PC のディスプレイの中へ移動でき る訳ではない。ここで VIWORD を使えば、ノート型 PC へのポインティングに加え、フ ァイルのコピーも簡単かつ直感的に行うことができる。

図 4-1 の上の図は、2 台の PC を結合した様子である。仮想ワークスペースビューを閉 じたら真中の図のように通常画面と変わらないように見える。しかし、右の PC でカーソ ルを左側の端へ持っていくと、左の PC の画面が表示され、下の図のようにマウス入力の リダイレクトが行われる。





図 4-1 VIWORD 利用例(一人のユー
------------------------

## 4.2 多人数での利用例

VIWORD の目的は一人で使うことではなく、共同作業を行うグループを支援するため である。他のユーザが何をやっているか、作業がどこまで進んでいるかなどの状況を知る ためにその人の作業机まで行かなくても、VIWORD の仮想ワークスペースビューを開く だけで良い。また、作業中にわからないことがあったら、他のユーザのデスクトップ上に 自分のデスクトップ画面を転送して、代わりに操作してもらうことも可能である。

他のコンピュータを遠隔操作することは、VNC [15] や Windows のリモートデスクトッ プ <sup>†13</sup> などを使っても可能である。しかし、自分のコンピュータにサーバアプリケーショ ンを起動しておいて、他のユーザに IP アドレスとパスワードを伝えなければならないこ とは煩雑である。これを、VIWORD を用いてば簡単かつ直感的に行うことができる。仮 想ワークスペース上で、転送元のピアパネルを転送先のピアパネルへドラッグし、レプリ カパネルを生成してあげれば良い。すると、転送先のデスクトップ画面上にレプリカウィ ンドウが現れるので、その中で遠隔操作してもらうことができる(図 4-2)。



図 4-2 VIWORD 利用例 (多人数のユーザ)

 $<sup>^{+13}</sup>$ 

http://www.microsoft.com/japan/windowsxp/pro/business/feature/remote/remotedesktop.ms px, accessed January 2009

### 4.3 共用画面を含む利用例

共同作業を行う際に共用画面を用いることが多い。例えば、ミーティングを行うとき、 プロジェクタに一つのコンピュータをつなげて、大きなスクリーンに投影された画面をグ ループのみんなで見ながら議論を進めることがよくある。しかし、一般の共用画面は同時 に一台のコンピュータとしかつながらず、他のコンピュータの画面を共用画面に投影する ためにはディスプレイケーブルをつなぎ直さなければならない。さらに、共用画面を他の ユーザが操作することも不可能である。ここで VIWORD を使えば、ディスプレイケーブ ルをつなぎ直さなくても共用画面に投影する画面の切り替えを簡単かつ直感的に行うこと がで、さらに、他のコンピュータから共用画面を遠隔操作することもできる。

図 4-3 では、2 台の PC と共用画面一つが VIWORD を利用している。二枚目の図では、 レプリカパネルを生成して、左の PC のデスクトップ画面を共用画面へ転送を行っている。 この場合のように、画面転送を左の PC (転送元) でなく右の PC (第三者) が行うことも 可能である。また、三枚目の図のように、共用画面上に複数のデスクトップ画面を表示さ せることもできる。さらに、共用画面をユーザ PC へ転送することで、ユーザが自分の PC で共用画面を遠隔操作することも可能である。もちろん、下の図のように、ユーザ PC と 共用画面を結合させても遠隔操作が可能である。







図 4-3 VIWORD 利用例(共用画面あり)

## 4.4 複数の画面を結合して大画面を構成する利用例

VIWORD の画面結合機能を用いれば、小さいディスプレイを組み合わせて大画面を構成することも可能である。その例を図 4-4 で示す。まずは、上の図のように、5 台の PC で VIWORD を起動させ、そのうち 4 台を結合させる。そして、真中の図のように、5 台目の PC のデスクトップ画面を結合された画面へ転送する。このとき生成されたレプリカパネルとレプリカウィンドウの位置は自由に変更できる。さらに、下の図のように、レプリカパネルをダブルクリックしてフルサイズモードにすると、レプリカパネルとレプリカウィンドウが、結合された4つのディスプレイ全体を覆うようになる。







図 4-4 VIWORD 利用例(結合による大画面構成)

## 第5章 関連研究

Weiser は現在のコンピュータのあり方を否定し、その究極に進化した姿は環境に統合さ れてユーザはそれをコンピュータと認識せずにその恩恵を受けるものであるといった**ユビ** キタスコンピューティング (Ubiquitous Computing) を提唱した [11] 。そして、コンピュ ータのユビキタス化はサイズによって、inch-scale の TAB と foot-scale の PAD と yard-scaleのBOARDの3つに分類されるという。ユビキタスコンピューティングの提唱 から約20年が経った今、RFIDタグが広く使われ始め、タブレットPCも普及し、様々な 場所で大画面が置かれるようになった。少しずつではあるが Weiser の夢見た世界へと進 んでいるといえる。少し視点を変えて見れば、TAB はともかく PAD と BOARD は結局の ところディスプレイの形をすることがわかる。複数のユーザが複数のコンピュータ(ディ スプレイ)を用いて自然に共同作業を行える未来を描いているが、少なくとも現時点での ユーザ達はコンピュータのパーソナライズ化を諦めようとしない。というより、まだその 基盤技術が整っていないと言った方が正確かもしれない。このような現状で今注目すべき 問題点は、パーソナライズ化したディスプレイ(PAD サイズ)と共用ディスプレイ(BOARD サイズ)が混在する中、ディスプレイ間の情報共有を直感的な方法で行うことが可能なユ ーザインタフェースであると考えられる。Streitz らは i-LAND [14] と呼ぶインタラクテ ィブルームのコンセプトを実現したデバイスをいくつか紹介したが、その中で PAD の画 面を BOARD の中に投影するようなものがある。それは、椅子とノート型 PC を組み合わ せた CommChair の操作画面を、壁面インタラクティブディスプレイ DynaWall へ転送す ることである。

TeamWorkStation [4] は自分のコンピュータのデスクトップ画面に他のコンピュータ のデスクトップ画面やカメラ画像を半透明にオーバーラップ表示させることで、互いの作 業領域に対しての指さしや間接的な描画を可能にした。しかし、同一の対象物を見ること はできても、それを直接操作することは不可能であり、ビデオ会議を拡張したものに過ぎ ないといえる。転送されたデスクトップ画面を遠隔操作するシステムとしては VNC [15] が挙げられる。しかし VNC は、見る側(クライアント)が見られる側(サーバ)を指定 して接続するように設計されているため、自分のデスクトップ画面を積極的に他のコンピ ュータへ転送することができない。共同作業を行うグループの支援というより、一人のユ ーザによる複数のコンピュータの利用を支援することが目的である。

Mighty Mouse [9] は VNC を利用したシステムであり、複数のコンピュータの中から、 遠隔操作を行う対象コンピュータを切り替えるための、インタフェースを二つ提供してい る。その一つは、コントロールパネルから操作対象を選択することであり、もう一つは、 ディスプレイの両端にカーソルを置いたときに、コントロールパネルの並び順に従ったコ ンピュータの画面に切り替わることである。同じく VNC を利用するシステムである comDesk [13] は、コントロールパネルに表示される各コンピュータのデスクトップ画面 のサムネールをドラッグして画面転送を開始する。そして、画面転送が行われている状況 がわかるように矢印を表示している。

Tee らは Community Bar と呼ぶインスタントメッセンジャーを拡張し、デスクトップ 画面転送・遠隔操作の機能を盛り込んだ [10] 。作業状況のアウェアネスとプライバシー のトレースオフを考慮し、転送する領域やズームレベルの最大値を設定できるようにした 上、ぼやけなどのマスクのフィルタを施すことを許容している。Community Bar は元々 離れた場所にいるユーザ同士での共同作業を支援することに焦点を置いているため、ビデ オチャット機能も備えてあることが特徴といえる。ただし、それと同じ理由で、共用画面 へ向かっての画面転送は考慮されなかった。同じ場所にいるユーザ同士での共同作業を支 援するシステムに TEAM STORM [7] というシステムがある。このシステムは、デザイナ グループのアイディア会議を支援することを目的としている。各自のユーザの持つタブレ ット PC に描いたデザインを共用画面へ転送し、他のユーザと共有できる。スケッチ専用 ではあるが、共同作業と個人作業を並行して行えることは高く評価できる。

Biehl らによる Impromptu [6] は、あらゆるアプリケーションを共有するために、任意 のアプリケーションウィンドウのピクセルデータを転送する方法を採択した。アプリケー ションウィンドウごとに Show と Share モードを選択可能にし、他のユーザのアプリケー ションウィンドウの複製を自分のデスクトップ上に表示することを許容する。さらに、共 有画面を共同作業に加えることができる。共用画面には、複製ウィンドウの配置やキーボ ード・マウスのリダイレクションが可能である。Tan らの WinCut [3] は、アプリケーシ ョンウィンドウの中の任意の領域を複製して独立したウィンドウにするシステムである。 WinCut は別のコンピュータへ移動させることができ、共用画面に複数のユーザから送ら れた WinCut を配置させ、同時に見比べることも可能である。ただし、共用画面を操作す るためには、別途の入力リダイレクトシステムを利用しなければならない。

Biehl らが Impromptu より前に開発した ARIS [5] は、インタラクティブスペース内の 共用画面やノート型 PC、PDA などで稼働しているアプリケーションを他のデバイスへ移 動させることができる。ARIS の特徴は、ディスプレイの物理的な位置関係に対応するマ ップインタフェースを提供するところにある。ユーザはそのマップインタフェースを用い て、マウスドラッグによる直感的な方法で、アプリケーションの移動を行うことができる。 Johanson らは iRoom [1] と呼ぶインタラクティブスペースを構築し、5 つの組み込まれ たディスプレイ (4 つは壁面、1 つはテーブルトップ)とカメラやマイク、ユーザ持ち込 みの各種デバイス (PDA、ノート型 PC など)間でのデータ交換と遠隔操作を可能にした。 そして PointRight [2] と呼ぶキーボードとマウスの入力をリダイレクトするシステムを 取り入れ、どのマウスを用いても(iRoom 専用/ユーザ持ち込み)ディスプレイ間の物理 的な境界を跨いだカーソルの移動を可能にした。これらのシステムの限界は、特定の環境 でしか使えず、ディスプレイ間のトポロジーをユーザ自身が必要に応じてダイナミックに 変更できない点である。 E-conic [12] は複数のディスプレイ間でカーソルやウィンドウをユーザの視点に合わせ て移動させることができる。ディスプレイは窓ガラスみたいなものに見立て、ユーザがそ れを通して仮想平面を見るような感覚を覚えさせることを目的とする。ウィンドウはユー ザの立ち位置に合わせてサイズと形を変えることで、ユーザに対して常に垂直にあるよう に見せる。複数のユーザがいる場合はウィンドウを誰の視線に合わせるかを選択できる。 カーソルはディスプレイの間を飛び移るのではなく、ディスプレイの間の映らない領域に あることもある。カーソルやオブジェクトをディスプレイの境界を越えて移動させること は、Rekimotoらによる Augmented Surfaces [8] で紹介された Hyperdragging と呼ぶ手 法でも行われた。ユーザは、テーブルの上置いたノート型 PC からオブジェクトをドラッ グして他のノート型 PC やテーブルの上、壁面に移動させたり、テーブル上の物理的なオ ブジェクトをコンピュータに取り入れたりすることができる。

HaらはSwordfish [16] というディスプレイ間の連結を簡単に行うためのフレームワー クを開発し、それ利用して作成したインタフェースをいくつか紹介した。その中で、ディ スプレイ間の連結をグラフィカルに表示するマップインタフェースは、マウスのドラッグ &ドロップ操作により、ディスプレイ間の連結を生成・変更することを許容する。ディス プレイを表すアイコン周辺の8つの点(上下左右+角)の一つと、他のディスプレイアイ コンの周辺の点の一つを結ぶことで、ディスプレイ同士が連結される。ディスプレイの連 結はユーザ間で共有されず、各ユーザは各自の設定を保持する。

## 第6章 議論

### 6.1 本研究の位置付け

コンピュータという機械にとって、その中核に当たるものはプロセッサやメモリなどの 電子部品であるといえる。しかし、それらの電子部品は普通本体の中に隠れていてユーザ に見えない、むしろ見せる必要のないものである。ユーザに見せる必要があるのはディス プレイ、キーボード、マウス程度のユーザインタフェースを担当するものである。さらに、 ユーザはキーボードとマウス自体に対して操作を行うのでなく、画面に向かっての操作に それらを用いているだけである。以上のことから、ユーザにとってコンピュータの中核に 当たるのは、他ならぬディスプレイであるといえる。ユーザがコンピュータを使用すると き、画面に対して何かの操作をして、その結果で変化した画面を見て何らかの情報を得る、 といった一連の流れが繰り返されることになる。多くの場合、コンピュータを操作すると いうことは画面を操作することを指す。

ー台のコンピュータに複数のディスプレイを使用(マルチディスプレイ)した体験のあ るユーザは、ある不思議な現象を普通のことのように認識してしまう。それは、マウスカ ーソルが画面の物理的な境界を通り抜けて別のディスプレイに移動することである。さら に、マウスのドラッグ操作により、アイコンやウィンドウをも別のディスプレイに移動さ せることもできる。一方で、二台のコンピュータがそれぞれの一面のディスプレイを備え ていて、そのディスプレイ同士を隣り合わせた場合を想像してみよう。マルチディスプレ イを使用した経験がないユーザならともかく、マルチディスプレイに手慣れたユーザなら、 カーソルが両方の画面を行き来できないことにストレスを感じる。また、画面間でファイ ルを移動させようとしたら、USBメモリのような補助記憶装置を用いるか、FTP などネ ットワークを経由しての方法を取るしかない。どちらにしても、ユーザはコンピュータ本 体を強く意識せざるを得ない。

今度は一人のユーザではなく、オフィスや研究室などの場において共同作業(Collocated Collaboration)を行うグループを考えよう。このような場合の一般的な問題点の一つは、 各ユーザは他のユーザの作業状況を把握しにくいということである。そして、その一般的 な解決法は、他の人の机までに歩いて行ってそのユーザの画面を見ながら会話することで ある。また、作業の途中で何かわからなくなって、自分のコンピュータを他の人に操作し てもらいたい場合がある。この場合の一般的な解決法は、相手を自分の作業机に来てもら うことである。

このような煩雑な身動きをなくす方法として、VNC のようなリモートデスクトップシ ステムを利用すれば、場所の移動なく他のユーザのデスクトップ画面を閲覧・操作を行う ことができる。しかし、デスクトップ画面を転送する側はサーバアプリケーションを起動 した後に、遠隔操作するユーザにネットワークアドレスを伝える必要があるため、根本的 な解決にはならない。

さらに、同じ場所での共同作業には共用画面が欠かせない。ミーティングルームや教室 などではほとんど共用画面が設置されていて、オフィスや研究室などでも共用画面を導入 されつつある。このようなたくさんのディスプレイが混在している環境(MDE, Multi-Display Environments)の中で、いかにして共同作業を行うグループを支援するかは重要な課題で ある。

#### 6.2 本研究の貢献

ユーザがコンピュータを用いて作業するとき、その作業の対象として認識するのはディ スプレイに表示されるデスクトップ画面ある。したがって、複数のユーザが共同作業を行 う場合も、各ユーザのデスクトップ画面こそがその共同作業の主体であり、媒体になるべ きである。そのために、共同作業を行うグループのすべてのユーザのデスクトップ画面を 仮想ワークスペースに配置し、すべてのユーザがいつでも他のユーザの作業状況を見られ るようにした。仮想ワークスペースの中で各デスクトップ画面自体が操作の対象として扱 うことができる。すなわち、デスクトップ画面を他のデスクトップ画面の上に重ねたり、 デスクトップ画面同士を結合させたりすることが可能である。さらに、ユーザはファイル を共有するにもコンピュータを意識する必要がない。仮想ワークスペース上に公開したフ ァイルを"画面"に対して転送を行うことが可能である。

### 6.3 今後の展望

仮想ワークスペースを実現した VIWORD は、現在の実装でも一人のユーザ、もしくは、 少人数のグループで十分実用的であるが、プロトタイプである故にいくつかの不完全なと ころがある。その中で一番の欠点として画面転送における遅延時間が挙げられる。デスク トップ画面を小さな領域に分けて PNG 圧縮をするなどの通信量を減らす努力はしている が、画像処理にかかる時間も無視できないため、それ以上の高度な処置は難しい。これは トレードオフの問題であり、実は画面転送による共同作業支援システムの共通の課題でも ある。一つ希望を持っているのは、将来にはプロセッサやネットワークの発展により大き な問題でなくなることである。

また、デスクトップ画面の結合機能も不完全である。結合されたディスプレイの上に他 のデスクトップ画面を転送することはできても、一つのディスプレイに結合されたデスク トップ画面を転送することはできないのが現状である。さらに、転送するデスクトップ画 面のサイズを変更する手段がフルサイズモード以外にないことも、ユーザにとって大きな 制約である。今後このような問題を解決し、より自由度の高い操作が可能になるように仮 想ワークスペースビューを改善する必要がある。 現在のVIWORDでは、ファイルを仮想ワークスペース上に公開することが自分のデス クトップ画面でしか行うことができない。すなわち、遠隔操作中のデスクトップ画面から ファイルを転送できない。セキュリティのためならそれで良いかも知れないが、ネットワ ークレベルで保護されているグループにとっては不便なことである。特に、共用画面をよ り有用なものにするためには、共用画面を遠隔操作して作業を行った成果物のファイルを 共有したい要望が高い。さらに、結合されたディスプレイ同士でファイルをドラッグ&ド ロップで移動できないことは、マルチディスプレイを使用する場合と大きな差がある。カ ーソルの移動だけでは、マルチディスプレイのような操作ができるとはいえない。今後こ の直感的な操作を実装し、VIWORD のユーザビリティを高めていきたい。

さらに、仮想ワークスペースを電子掲示板(BBS, Bulletin Board System)のように活 用することも考えられる。例えば、簡単なテキストや画像のメモを残すことで、共同作業 に関連する情報を他のユーザ達に非同期的に伝えることができる。同様に、デスクトップ 画面の一部を指定して切り取った画像も仮想ワークスペースに配置することも検討してい る。

## 第7章 結論

本研究では、複数台のコンピュータのデスクトップ画面を仮想ワークスペースに集め、 それらの包含関係、結合関係をダイナミックに変更できるインタフェースを提案し、プロ トタイプである VIWORD の実装を行った。

ユーザは仮想ワークスペースビューを通じて他のユーザの作業状況を確認することがで きる。また、マウスのドラッグ&ドロップによる直感的な操作で、デスクトップ画面の転 送や結合を行うことができる。転送されたデスクトップ画面上でのマウスとキーボードの 入力は、転送元のコンピュータへリダイレクトされる。結合されたデスクトップ画面はま るでひとつの画面であるかのように扱うことができ、ディスプレイ間でのカーソルの移動 も可能である。さらに、仮想ワークスペース上にファイルを公開し、他のユーザのデスク トップ画面へ転送することができる。一人のユーザが仮想ワークスペースを操作すると、 すべてのユーザの仮想ワークスペースが同期して更新される。

以上の機能は、ユーザにコンピュータを意識させず、画面自体を対象とする直感的な操 作で実現できる。また、特定の場所に留まらず、予期せぬ共同作業までをも支援するため、 サーバを要しない P2P 方式での実装を行っている。VIWORD は起動と同時にローカルネ ットワーク上で稼働中の VIWORD を探索して直ちに共同作業に参加する。

VIWORD は、共用画面を含む共同作業の支援を主な目的とするが、個人で複数台のコンピュータを利用する場合でも有用である。

## 謝辞

本研究を行うにあたり、そして本稿の執筆にあたり、ご指導頂きました田中二郎教授、 ならびに高橋伸講師、三末和男准教授、志築文太郎講師に心より感謝申し上げます。また、 色々な意見を頂きましたインタラクティブプログラミング研究室の皆様、特にユビキタス チームのメンバの方々に感謝の意を表します。

最後に、いつも支えてくれる家族および友人にも改めて感謝致します。

## 参考文献

[1] Brad Johanson, Armando Fox, Terry Winograd, The Interactive Workspaces Project: Experiences with Ubiquitous Computing Rooms, IEEE Pervasive Computing, v.1, n.2, p.67-74, April 2002

[2] Brad Johanson, Greg Hutchins, Terry Winograd, Maureen Stone, PointRight: Experience with Flexible Input Redirection in Interactive Workspaces, Proceedings of the 15th annual ACM symposium on User interface software and technology, p.227-234, October 27-30, 2002

[3] Desney S. Tan, Brian Meyers, Mary Czerwinski, WinCuts: Manipulating Arbitrary Window Regions for More Effective Use of Screen Space, CHI '04 extended abstracts on Human factors in computing systems, p.1525-1528, April 24-29, 2004

[4] H. Ishii, TeamWorkStation: Towards a Seamless Shared Workspace, Proceedings of the 1990 ACM conference on Computer-supported cooperative work, p.13-26, October 07-10, 1990

[5] Jacob T. Biehl, Brian P. Bailey, ARIS: An Interface for Application Relocation in an Interactive Space, Proceedings of the 2004 conference on Graphics interface, p.107-116, May 17-19, 2004

[6] Jacob T. Biehl, William T. Baker, Brian P. Bailey, Desney S. Tan, Kori M. Inkpen, Mary Czerwinski, Impromptu: A New Interaction Framework for Supporting Collaboration in Multiple Display Environments and Its Field Evaluation for Co-located Software Development, Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems, p.939-948, April 05-10, 2008

[7] Joshua Hailpern, Erik Hinterbichler, Caryn Leppert, Damon Cook, Brian P. Bailey, TEAM STORM: Demonstrating an Interaction Model for Working with Multiple Ideas During Creative Group Work, Proceedings of the 6th ACM SIGCHI conference on Creativity & cognition, p.193-202, June 13-15, 2007

[8] Jun Rekimoto, Augmented Surfaces: A Spatially Continuous Work Space for Hybrid Computing Environments, Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit, p.378-385, May 15-20, 1999

[9] Kellogg S. Booth, Brian D. Fisher, Chi Jui Raymond Lin, Ritchie Argue, The "Mighty Mouse" Multi-Screen Collaboration Tool, Proceedings of the 15th annual ACM symposium on User interface software and technology, October 27-30, 2002

[10] Kimberly Tee, Saul Greenberg, Carl Gutwin, Providing Artifact Awareness to a Distributed Group through Screen Sharing, Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work, p.99-108, November 04-08, 2006

[11] Mark Weiser, The Computer for the 21st Century, Scientific American, p.94-100, September 1991

[12] Miguel A. Nacenta, Satoshi Sakurai, Tokuo Yamaguchi, Yohei Miki, Yuichi Itoh, Yoshifumi Kitamura, Sriram Subramanian, Carl Gutwin, E-conic: a Perspective-Aware Interface for Multi-Display Environments, Proceedings of the 20th annual ACM symposium on User interface software and technology, p.279-288, October 07-10, 2007

[13] Motoki Miura, Buntarou Shizuki, Jiro Tanaka, comDesk: A Cooperative Assistance Tool based on P2P Techniques, Proceedings of 8th International Conference on Knowledge-Based Intelligent Information and Engineering Systems, v.2, p.883-890, October 2004

[14] Norbert A. Streitz, Jörg Geißler, Torsten Holmer, Shin'ichi Konomi, Christian Müller-Tomfelde, Wolfgang Reischl, Petra Rexroth, Peter Seitz, Ralf Steinmetz, i-LAND: An interactive Landscape for Creativity and Innovation, Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit, p.120-127, May 15-20, 1999

[15] Tristan Richardson, Quentin Stafford-Fraser, Kenneth R. Wood, Andy Hopper, Virtual Network Computing, IEEE Internet Computing, v.2, n.1, p.33-38, January 1998

[16] Vicki Ha, Kori Inkpen, Jim Wallace, Ryder Ziola, Swordfish: User Tailored Workspaces in Multi-Display Environments, CHI '06 extended abstracts on Human factors in computing systems, p.1487-1492, April 22-27, 2006