

# ネットサーフィンするマウスカーソル – マルチ計算機マルチマウスシステムの開発 –

上田 真史

masa-u@nue.ci.i.u-tokyo.ac.jp  
東京大学大学院情報理工学系研究科

竹内 郁雄

nue@nue.org  
東京大学情報理工学系研究科

## 概要

マルチマウスシステムは、簡易で安価ながらマルチユーザシステムを構築できるなど、非常に応用性の高いシステムである。しかしながら、特にユーザ数を増やした場合に作業スペースが足りなくなりやすいという問題もかかえる。この問題を解決するために、昨年開発したマルチマウスミドルウェア MMTk を拡張し、ネットワークを通じマウスカーソルが移動できる機構を実装した。この拡張により、より大きな作業スペースを持つマルチマウスシステムを実現できるほか、新たな応用も生まれるだろう。

## 1 はじめに

PCの標準的入力デバイスであるマウスは、目下1台のPCに1台のみ接続するというスタイルが定着してしまっている。しかし、USBマウスが安価で市場に溢れていて、PCに複数のマウスを繋ぐのは簡単である。複数のマウスをPCにつないで活用できれば、マウスやPCの活躍する範囲をもっと広げられるだろう

— という思想をもとに、我々はマルチマウスシステムの開発を行っている。これまでに、Windows上で動作するマルチマウスミドルウェアとしてMMTkを開発しwebにて公開している。また、MMTkの上で実際に動作するアプリケーションソフトウェアとして、災害避難所用情報収集システムやマルチプレイヤーマインスイーパーを開発した。

今回はこのMMTkをさらに拡張させ、よりフレキシブルな応用が可能となるよう現在開発を行っている、ネットワークを採り入れた新しいMMTkの機能について紹介する。

## 2 マルチマウスシステムについて

マルチマウスシステムとは、その名のとおりに、マウスを複数つないだシステムのことであり(図1)。マルチマウスシステムの最大の特徴であり

特長である点は、なんとといっても、同一のワークスペース上で複数のポインティングデバイスを扱えることである。また、ごく一般的なPC1セットと複数のマウスがあれば構築できるので、PCが既に存在していればあとは数千円の投資でマルチマウスシステムを構築できるという、抜群のコストパフォーマンスも備えている。

マルチマウスシステムのSingle Display Groupware (SDG) [1]としての応用が近年特に注目されている。University of Marylandでは、教育現場において生徒同士の協調作業をサポートするためのKidPadというアプリケーションソフトウェアの開発を通じ、SDGが研究されている[2]。Microsoftでは、発展途上国の教育現場において生徒の人数分の計算機を購入することができない場合に向け、複数の生徒に1台の計算機を使わせるというスタイルが研究されている[3]。これらはCSCWの一種とみることができる。

以上は1人が1台のマウスを使うことにより1台の計算機を複数人で使用するシステムであるが、もちろん1人が複数のマウスを同時に扱ってもよい。大阪大学では、1人が2台のマウスを使い両手を同時に動かすことで新たなインタフェースを創造する、ダブルマウスシステム[4]が研究されている。

Mouse Cursors Surf the Net – Developing Multi-computer Multi-mouse Systems  
Masafumi Ueda and Ikuo Takeuchi, Graduate School of Information Science and Technology, The University of Tokyo

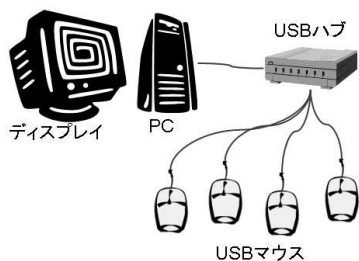


図 1: マルチマウスシステム

## 2.1 我々のアプローチ

我々は、マルチマウスシステムを災害対応の現場に応用することをきっかけに、マルチマウスシステムの開発をはじめた。

災害対応の現場においては、複数の人が1枚の紙、特に紙地図を取り囲み、一齐に作業を行うことが多い。この作業を電子化し、作業の効率化を狙うとともに、情報の共有を促進しようというものである。実際に災害避難所用情報収集システム(図2)を開発し実験を行い、43件の情報を5分で入力できたなど、マルチマウスシステムの小規模かつ高効率な点を確かめることができた [5]。

## 3 MMTk について

MMTk は、我々が一昨年より開発を行っている、マルチマウスシステム開発用のミドルウェアである<sup>1</sup>。上で述べた避難所用システムも、この MMTk の上に実装されている。

MMTk は Windows XP の Raw Input API を使い、マウスからの情報をマウスごとに分別し、イベントコールバックという形でアプリケーションプログラムに提供する。またマウスカーソルもマウスの台数分描画し、アプリケーションプログラムのマルチマウス化にかかわる開発の負担を軽減する。イベントコールバックは Windows の従来からのウィンドウメッセージに似た形とし、Windows プログラミングに慣れた開発者が

<sup>1</sup>MMTk は、Multi-Mouse Toolkit とも、みんなでマウス使おうミドルウェアとも。



図 2: 災害避難所用システムの画面

容易にマルチマウスアプリケーションを開発できるようにしてある。

MMTk は Windows の API を使用し、特に専用のデバイスドライバ等は必要としない。アプリケーションプログラムのダブルクリックのみでマルチマウス環境がスタートするのも、MMTk の大きな特長である。

MMTk は現在、開発者用パッケージを一般公開しており、ホームページ<sup>2</sup> からダウンロードできる。現在いくつか応用に関する問い合わせがきており、今後さらなる応用がされるものと期待される。

## 4 拡張の内容

今回の拡張では、マウスカーソルをネットワークを通じて他の計算機へ移動できるようにする。これには、まずマルチマウス環境における作業領域の拡張という狙いがある。

マルチマウスシステムでは、人数が増えるとマウスカーソルの数が増えるとともに、必要な作業領域もぐんぐん増えてゆく。狭いディスプレイに多数のマウスポインタを配置していくと、いずれマウスポインタがディスプレイじゅうにひしめくことになる(図3)。狭いディスプレイを複数人で使用するとひとりあたりの作業スペースの解像度が低下し、まともに表示内容を見ることができないということも発生する。

<sup>2</sup><http://www.nue.ci.i.u-tokyo.ac.jp/~masa-u/mmtk/>

これは私の昨年の避難所システムの実験においても実際に発生した問題で、地図上にて7つのマウスの動作領域をとると地図上の物体が見えづらくなるという問題が発生した。たった7つのマウスの領域ですら発生する問題であるから、それ以上にマウスカーソルが増えた場合にこの問題がクリティカルになるのは明らかである。

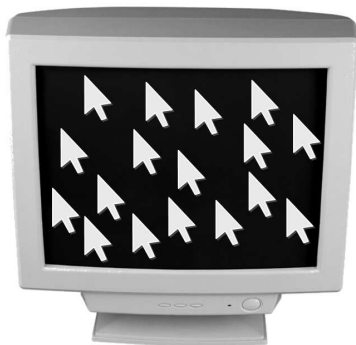


図 3: 過密なマウスカーソル

解決方法としてディスプレイを大きくすればよいと言うのは簡単であるが、現実的な問題として、ディスプレイの大きさ、すなわち解像度の高さには限界がある。もちろん、特殊なディスプレイを用いれば超解像度を得ることも難しくはないが、そういった機材はたいへんに高価である。マルチマウスシステムはマウスを用いることで安価に構築できるものである。そんなところに高価な機械を投入してしまうと、せっかくの特長を損ねてしまう。

マウスカーソルをネットワークを通じて移動可能とすれば、計算機複数台の画面スペースを結合した大きな領域でのマルチマウス環境が実現できる。計算機は特殊な大型ディスプレイよりずっと安価であり、一般的な PC を使うことができるので、マルチマウスシステムのコストエフェクティブな面も残すことができる。

#### 4.1 マウスカーソルをネットワーク経由で操作にすることについて

マウスの情報をネットワーク経由で他の計算機へ渡し、遠隔のマウスカーソルを操作すると

いうシステムは、主に計算機を遠隔操作する目的で多く使われている。VNC や Remote Desktop がそれにあたる。

マウスカーソルの位置に応じて操作する計算機を自動的に変えるシステムも存在する。x2vnc [6] や Synergy [7] といったシステムがそれである。Stanford University で研究されている PointRight [8] というシステムは 3 次元的なトポロジの構成も可能である。

しかしながら、いずれのシステムにおいても、1 台の計算機に同時に存在できるマウスカーソルの数は 1 つのみである。また操作対象が従来のアプリケーションプログラムであるため、結局スクリーンが違えば環境が変わってしまう。

MMTk ではこれらの制約を取り払い、完全にシームレスな、n 台の計算機に m 台のマウスを接続し運用できるシステムをめざす。

## 5 現在のプロトタイプシステム

以上をふまえ、現在、ネットワーク対応新 MMTk のプロトタイプシステムができあがっている。ここではこのシステムを仮に *i*MMTk と呼ぶ。現在のプロトタイプシステムはサーバクライアントシステムとして実装されている。

### 5.1 サーバ

サーバは複数の仮想マウスカーソルと、それが動きまわるための 1 つの仮想スクリーンを用意し、クライアントに提供する。

注意したいのは、仮想マウスカーソル、仮想スクリーン、ともにサーバは提供を行うのみで、その動作や内容についてはサーバが能動的に関わることはないという点である。これには、仮想マウスカーソルや仮想スクリーンが、サーバの持つ現実のマウスや現実のスクリーンとは無関係であるということも含む。

たとえば、マウスを全く持たないマシンをサーバとしてもよいし、現実のスクリーンより何倍も広い仮想スクリーンを持っていてもよい。もちろん、クライアントとサーバを同一の計算機上に配置することも可能である。

## 5.2 クライアント

一方でクライアントは現実のマウスとスクリーンを持っている（ことになっている）マシンである。

クライアントは（たいてい）現実のスクリーンおよび現実のマウスを、それぞれ仮想スクリーンと仮想マウスカーソルに対応させる。仮想スクリーンはクライアント側にも存在し、これはサーバに存在する仮想スクリーンのある領域と対応している（図4）。ただし、これは仮想スクリーンのグラフィック的内容が一致しているということではなく、単純に幾何学的に対応しているというだけである。

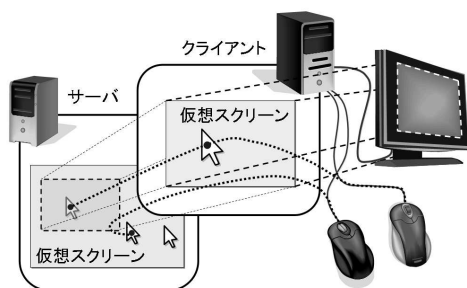


図4: 仮想オブジェクトと実オブジェクトとの対応

クライアントには一般のクライアントとオムニシエント・クライアントが存在する。どちらのタイプのクライアントになるかはクライアントが自ら申告することで決定する。オムニシエント・クライアントについては6.1.2節で説明する。

## 5.3 仮想マウスカーソル

マウスカーソルは仮想スクリーン上を動作する物体である。標準的な使いかたとしては、一般的なマウスカーソルと同様に、現実のマウス1台とマウスカーソル1つを対応させて、マウスの動作をマウスカーソルの動作に反映させる。他の使いかたでは、対応するマウスのない「真の」仮想マウスカーソルが挙げられる。対応するマウスのない場合は、その仮想マウスカーソルはプログラムが制御することになる。

## 5.4 仮想スクリーン

仮想スクリーンはサーバ側にあるものとクライアント側にあるものとの少々機能が異なる。サーバ側の仮想スクリーンは仮想マウスカーソルが動きまわるための仮想的な平面であり、そのサーバに存在するマウスカーソルのための、いわば「世界」である。

一方、クライアント側の仮想スクリーンは、サーバ側の仮想スクリーンの一部分または全部にアクセスするためのインターフェースとして使用される。サーバ側の仮想スクリーンはプロジェクタによって大きな壁に投影された画像、クライアント側の仮想スクリーンはその壁に貼られている画用紙、と考えていただきたい。仮想マウスカーソルがこの画用紙の中に移動し、画用紙に仮想マウスカーソルが映ることすなわち、仮想マウスカーソルがクライアントの仮想スクリーンに現れ、そして現実の画面に映るということである。

5.2節でも触れたが、サーバ側の仮想スクリーンとクライアント側の仮想スクリーンでは必ずしもグラフィック的内容が一致していない。とはいえ、上記プロジェクタの例から考えられるように、クライアント側の仮想スクリーンにサーバ側の仮想スクリーンから画像を「投影」したいこともあるだろう。そのようなことも実現できるように、仮想スクリーン間で描画命令を出したり、画像を転送する機能も実装した。

## 6 使いかた

まず、*vMMTk* の最も基本的な使いかたを説明する。

システムの外観は図5のようなものである。クライアントが3台存在し、それぞれに1台ずつのモニタと何台かのマウスが接続されている。このシステムは、ソフトウェア的には図6のようになっている。

サーバ上の仮想スクリーンは、クライアントのモニタの解像度を横に接続するかたちで合計した解像度を持ち、これを3分割したそれぞれの領域が、それぞれのクライアントの仮想スクリーンと対応している。



図 5: μMMTk のシステム外観

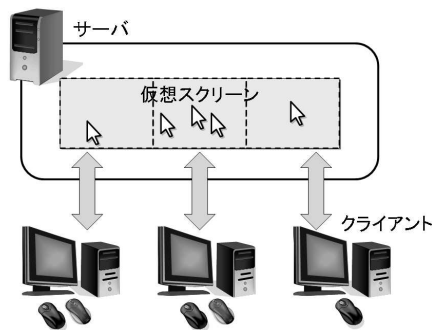


図 6: 図 5 のソフトウェア内部表現

クライアントのマウスを動かすとサーバの仮想スクリーン上で仮想マウスカーソルが動く。仮想マウスカーソルが動くにしたがい、仮想マウスカーソルは3つの領域間を移動する。ユーザが目にする仮想マウスカーソルのカーソル画像は、そのとき仮想カーソルが存在している領域に対応したクライアントのモニタ画面に描画される。よって、あたかもマウスカーソルがネットワークを通じて複数のモニタ画面を渡り歩いているように見える<sup>3</sup>。

マウスが動作した結果イベントが発生した場合は、そのとき仮想カーソルが存在している領域に対応したクライアントに送信される。クライアントはこれを受け取り、適切に処理する。

<sup>3</sup>これを「ネットサーフィン」というには LAN 内だとちょっと狭そうだ

## 6.1 クライアントの連携

さて、以上のようにして接続されたクライアント同士であるが、そのクライアント同士がどう動作するかにより、システム全体の活用法が変わってくる。

### 6.1.1 全く連携しない場合

クライアント同士が全く連携せずに、単に自分のところにやってくるマウスカーソルを受け入れる、というのは最も簡単なモデルである。このモデルは 4.1 節で述べた VNC のマルチマウスシステム版、ということになる。

このモデルは、ひとつひとつのアプリケーションプログラムは特に大きな作業領域を必要とせず、しかし様々なアプリケーションプログラムを多数動作させたい場合などに適しているだろう。

### 6.1.2 統括クライアントを置く場合

全部のクライアントを統括するスーパーバイザ的存在を置くのも、またひとつのモデルである。現在のプロトタイプシステムでは、このようなモデルのために、オムニシエント・クライアントというものを用意している。

オムニシエント・クライアントはいわば全智型のクライアントである。一般のクライアントが自分の仮想スクリーン領域とそこに入った仮想マウスカーソルしか把握できないのに対し、オムニシエント・クライアントはサーバの仮想スクリーン全域と、全ての仮想マウスカーソルを把握できる。

クライアントはサーバの仮想スクリーンに対して描画要求を出せるようにしてある。オムニシエント・クライアントがサーバに描画要求を出し、サーバがそれをさらに各クライアントに回送し、各クライアントが描画を行えば、オムニシエント・クライアントからはあたかも非常に大きな1枚のスクリーンを扱っているように見える。

このモデルは、システム全体でひとつの大きなアプリケーションプログラムを実行させたいときに適している。しかしながらシステムがあまりに大きくなると、1台のクライアントに処

理を集中させることの限界が見えてくるだろう。そこで次のモデルを挙げる。

### 6.1.3 分散して連携する場合

究極的には、クライアント同士が peer to peer で連携するのが望ましいだろう。マウスカーソルは全クライアントにわたってまんべんなく移動するものだとすれば、peer to peer で動作に関する情報を交換し処理するのが負荷の面でも最も望ましい。

vMMTk には、peer to peer での接続を支援する機構として、スクリーンやマウスカーソルの ID からクライアントのアドレスを引いたり、スクリーンやマウスカーソルの ID を使ってデータ送信を行えるような機能を備えている。

## 6.2 モデルの混在したシステム

vMMTk では、以上のクライアント連携モデルのいずれについても、システム全体の動作モデルをそれに固定することはしない。クライアントのうち一部は独立して動作し、また他のクライアントはオムニシエント・クライアントを中心に動作し…という具合であっても構わないというスタンスである。オムニシエント・クライアントが複数存在しても構わない。

図7はこの一例である。この図において、ディスプレイ1とディスプレイ2は、横に接続された状態で動作している。それぞれのクライアントである PC A と PC B は、特に連携せず独立に動いている。一方 PC C はオムニシエント・クライアントであり、仮想スクリーン全域が見えるが、この図においてはディスプレイ1とディスプレイ2を結合した領域だけについて注目している。

PC C はまた、ディスプレイ3において、ディスプレイ1の画面をウィンドウ内に表示している。

## 7 応用

vMMTk のシステムは、シングル計算機のマルチマウスシステムとはまた違った形態の環境

をユーザに提供する。この環境を活かす応用方法として、有用であろうものをいくつか挙げる。

### 7.1 大きな作業領域

これは当初の動機がそうであるから当たり前なのだが、vMMTk によりユーザは広大な作業領域を使えるようになる。いままで PC のディスプレイ1枚ではどうしても解像度が足りずにあきらめていた作業であっても、複数の PC を持ち寄って好きなだけ解像度の高いシステムを構築できる。

装置としての応用でまず考えられるのは、会議机である。テーブルトップにディスプレイを敷き詰めれば、かなり大きな領域の共同作業スペースを実現できる (図8)。

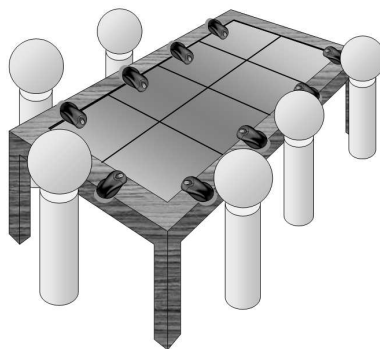


図8: 会議テーブルへの応用

vMMTk ではマルチスクリーンのうちどの部分をどう使うかという制御が柔軟にできる。会議机システムにおいては、どのような会議を行うかによって、全面を共有スペースとしたり、あるいは一部分を会議参加者が個人用に使うスペースに分割するなど、さまざまなスタイルの使用法を創出できる。余談であるが、このような机はゲームなどにも応用できそうである。

この柔軟性は災害対策本部や監視システムなどにおいても役に立つだろう。図9のように各係員が端末を持った状態で災害対策本部の大きな画面の前に集結したとき、各係員の端末からマウスカーソルを大画面へと移動させることができる。

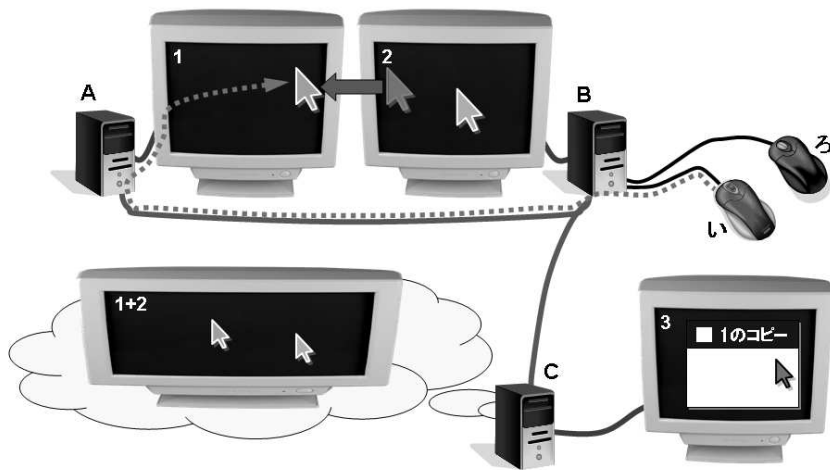


図 7: 様々なクライアントが存在している様子

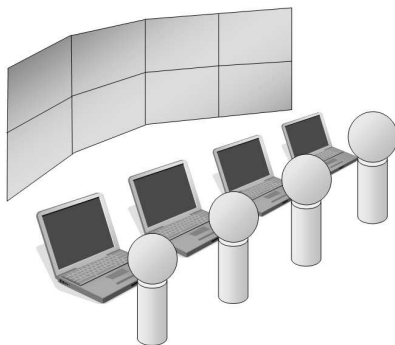


図 9: 災害対策本部への応用

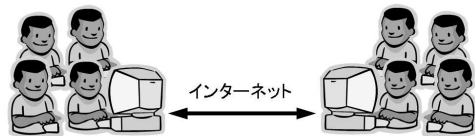


図 10: 接続されたマルチマウスシステム同士

ひとは独立してしまっている。マルチマウスシステム同士の接続はこれとは異なり、現実集合しているユーザグループ同士を接続するものである。

## 7.2 遠隔マルチマウスシステム

マルチマウスシステムはユーザが全員同じ場所に居るのが基本であるが、そこに遠隔のマルチマウスシステムもを繋げてしまうという用途にも、*μMMTk* は使用できる。たとえば遠隔の学校同士で共同作業を行う場合や、遠く離れたスポーツチーム同士が作戦練習を行うようなときに使用できるだろう。

従来より、ネットワークのある場所に皆が集合してなにかを行うというアプリケーションは存在しているが、そのいずれもが、各ユーザが端末を1台操作するものであり、ユーザひとり

## 7.3 さらに広げると

たとえば現在のプロトタイプシステムのサーバ同士が接続されると、仮想スクリーンのネットワークを構築することもできるだろう。こうすると仮想スクリーンを非2次元的に接続できるようにもなるので、どんどん仮想マウスカーソルの移動範囲を広げることができる<sup>4</sup>。

仮想マウスカーソルがどこにでも行けるようになると、インターネット上に広大な空間、すなわちサイバースペースが出現することにもなる。図11はそのうち一部のイメージで、カーソ

<sup>4</sup>ここからはちよつと妄想っぽくなりますが、おつきあい下さい。

ルがまさにネットサーフィンをする感覚で多数のディスプレイ上を動く。



図 11: マルチマウス・サイバースペース

その上で、ユーザひとりがマウスひとつを持っているとすれば、マウスカーソルひとつひとつを人格としてとらえることもできそうである。図中のマウスに手足を生やしてあるのは、これを表している。

このシステムを図 10 のようにインターネット上でつないでいくと、まさにサイバースペースとして、新しいソーシャルネットワーキングサービスなどに応用できないだろうか。地球の裏側の見えないモニタの画像をどうするか、トポロジをどう構築するかなど、実現には様々な課題があるが、未来のシステムとして挙げておきたい。

## 8 今後の開発

現在のシステムはあくまでプロトタイプであり、まだ一般公開できるようなものではない。今後一般公開できるよう、さらに開発を続ける。開発は、現在の MMTk のときと同様に、アプリケーションプログラムを並行して開発し、その結果をシステムの改良にフィードバックさせる形で行う。

## 9 まとめ

我々は以前よりマルチマウスシステムの開発を行っており、MMTk というマルチマウスシステム開発用ミドルウェアを公開している。昨年の研究においてマルチマウスシステムでは広大なスペースが必要であることがわかり、今回は

ネットワークを通じて PC を連携させることでワークスペースの拡大を行うよう、MMTk を拡張することにした。拡張後の MMTk システムは、仮想スクリーンと仮想マウスカーソルを用い、さらにオムニシエント・クライアント等の機能を導入し、ワークスペースをフレキシブルに構築できるようにした。

新しいシステムの応用としては、広大なスペースを利用する会議や災害対応などがまず考えられる。さらに、マルチマウスシステム同士を遠隔で接続するなどの利用法も考えられ、これを大規模化すれば何らかのサイバースペースをネットワーク上につくり出すこともできるようになるだろう。

様々な応用が実用になるよう、今後新しい MMTk の開発を進め、一般公開をめざす。

## 参考文献

- [1] Stewart, J., Bederson, B., Druin, A.: Single Display Groupware: A Model for Co-present Collaboration, *Proc ACM SIGCHI*, 1999
- [2] KidPad: <http://www.kidpad.org/>, University of Maryland
- [3] Pawar, U., Toyama, K., Pal, J.: Multiple Mice for Computers in Education in Developing Countries, *Intl. Conference on Information Technologies and Development*, 2006
- [4] ダブルマウスシステム: <http://www-nishio.ise.eng.osaka-u.ac.jp/tresearch/doublemouse/index.html>, 大阪大学大学院工学研究科 西尾研究室
- [5] 上田真史: 災害情報共有インタフェースおよび SDG ミドルウェアの開発, 電気通信大学大学院電気通信学研究科 修士論文, 2006
- [6] x2vnc: <http://fredrik.hubbe.net/x2vnc.html>, Fredrik Hubinette
- [7] Synergy: <http://synergy2.sourceforge.net/>, Syndergy Project
- [8] Johanson B., et. al.: PointRight: Experience with Flexible Input Redirection in Interactive Workspaces, *Proc UIST*, 2002