

## 複数ポインティング入力システムの構築

福地 健太郎<sup>†</sup> 松岡 聡<sup>†,††</sup>

複数のポインティング入力デバイスを同時に並行に操作可能なシステムをビデオカメラ入力ベースで構築した。一般的な GUI システムでは、画面上の複数の操作対象を並行に操作する事はできず、一つのポインティングデバイスを使って逐次的に処理する必要があった。本環境ではユーザーは最大 8 箇所まで同時に位置情報を入力する事ができ、GUI 画面上のオブジェクトを並行に操作する事ができる。各デバイスは透明なアクリル板の上で操作し、その様子は下からビデオカメラで撮影し、その画像から各デバイスの座標値を計算する。

### Implementation of Multiple Pointing Input System

KENTAROU FUKUCHI<sup>†</sup> and SATOSHI MATSUOKA<sup>†,††</sup>

This paper describes a prototype system of Multiple Pointing Input System(MPIS), which allows concurrent manipulation of multiple pointing devices. On the traditional GUI system the users have to manipulate each objects sequentially with one pointing device. Our MPIS allows users to point multiple places on GUI screen simultaneously, and manipulate multiple graphical objects (icon, slider) concurrently. The devices are manipulated on a clear acrylic table and the coordinate of each device is calculated from the images captured by a video camera below the table.

#### 1. Introduction

今日一般的な GUI システムでは、ユーザーはマウスやペン等のポインティングデバイスを用いて操作する。ユーザーはポインティングデバイスを用いてディスプレイ上の座標を入力し、画面上に表示された図形を選択・移動する。普通、ポインティングデバイスは一つであり、ユーザーは同時には一つのオブジェクトしか選択する事はできない。

**Multiple Pointing Input System(MPIS)** とは、ユーザーがコンピューター画面上の複数の座標を同時に指示する事ができるシステムである。これに対し、従来の GUI で一般的な、ユーザーはただ一箇所の座標を指示するシステムを、**Single Pointing Input System(SPIS)** と呼ぶ。

従来のアプリケーションにおいても、独立した操作が可能な操作対象が複数画面上に配置されていて、本来ならば並行操作の可能性がある場面は多い。例えばオーディオミキサーの GUI では、各音源のボリュームに対応したスライダーが表示されており、これらは独立に操作

できる。専用のミキシングコンソールではこれらのスライダーは並行に操作可能である。しかし、SPIS では、ユーザーは操作する対象を切り替えて逐次的に操作せざるを得ず、並行操作は不可能である。

本研究で目指す MPIS は、普通のデスクトップ機を使用するような環境で、指先で操作できるようなデバイスにより、こうした並行操作を可能にする汎用な入力システムである。また、複数の位置情報を用いた、より便利な入力手段をユーザーに提供する。本論文では、MPIS のプロトタイプ実装について説明し、またその上で構築した新しい入力技法やアプリケーションについて説明する。

#### 2. 関連研究

Tangible User Interface の一連の研究<sup>4)7)8)</sup> は、操作対象に強く関連付けられた入力デバイスを用意する事で、ユーザーがコンピューター上の操作対象をあたかも直接操作しているかのようなイメージを抱かせる事を目的としている。操作デバイスは Phicon と呼ばれ、その形状は、操作対象を強く想起させるような関連性の強いものになっている。例えば Illuminating Light<sup>7)</sup> では、ユーザーは光源・鏡・レンズの形状をした入力デバイスを机の上で操作する。しかしこうした Phicon を用いるインターフェースでは、入力デバイスはアプリケーションに特化したものになるため、複数のアプリケーション

<sup>†</sup> 東京工業大学情報理工学研究所数理・計算科学専攻  
Tokyo Institute of Technology, Graduate School of Information Science and Engineering, Dept. of Mathematical and Computing Science

<sup>††</sup> 科学技術振興財団  
Japan Science and Technology Corporation

を切り替えて使うデスクトップインターフェースには向かない。

Graspable UI<sup>2)</sup>は同時に8個のポインティングデバイスを操作できるシステムによる評価を行っている。評価実験では、画面上に4個の図形を表示し、それらを指定された場所・向きに合わせるタスクを課している。入力デバイスとして汎用な形状のものを使った場合と、画面上の図形に特化した外観を持つものを使った場合とを比較し、特化したデバイスを使う方が良い成績を出している。しかし、各入力デバイスはそれぞれに専用のタブレットの上でしか動かせない・各デバイスが大きく、同時に操作できるのは二個、といった不備がある。

ダブルマウス<sup>10)</sup>は、マウスを二個使用するインターフェースである。二つの操作対象を並行に操作する事が可能な事に加え、矩形範囲の指定では、矩形の端点を左右のマウスでそれぞれ指示すれば、一つのマウスで二点を指定するのに比べてより直感的であるとしている。また、アイコンの移動をする際に、その移動元と移動先をそれぞれのマウスで指示する事で、マウスの移動量を削減できるとしている。

Dual Touch<sup>9)</sup>は、既存の感圧式タッチパネルを用いて擬似的にMPISを実現するシステムである。ただし二箇所までしかポイントできず、また二つのポイント箇所を同時に動かす事はできない。

両手を用いた並行操作の有効性については、数多くの研究がある。Buxtonは両手による操作を、一つの対象に対する入力の種類を増す為に導入している<sup>1)</sup>。Kurtenbach<sup>5)</sup>やHinckley<sup>3)</sup>は一つのタスクを達成するための両手による協調動作の有効性を示している。

Pebbles<sup>6)</sup>では、複数のユーザーが一台ずつPalmPilotを使用して、共有された画面上にある各自のポインターを操作できる環境を構築している。一人のユーザーが複数のポインターを操作する事は想定していない。

### 3. Multiple Pointing Input System

Multiple Pointing Input Systemは、複数の位置情報を同時に入力し、GUI画面上の複数の箇所をポイントする事ができるシステムである。ユーザーは、両手の指先を用いて複数の位置情報を入力する。

#### 3.1 MPISによる操作

MPISが有効に働くと考えられるシナリオをいくつか挙げる。

**複数オブジェクトの同時操作** 画面上に複数の操作可能なオブジェクトがある場合、ユーザーはそれらのオブジェクトを独立にポイントする事で、同時に操作する事ができる。例えば、画面上の複数のファイルアイコンを、指先でポイントする事で同時にドラッグできる。先に挙げたオーディオミキサーの場合、複数のスライダーを並行に操作する事が可能となる。

**豊富なオブジェクト操作** 一つのオブジェクトに対し複数のポインティングデバイスを用いれば、複雑な指示をする事ができる。例えば二つのポインティングデバイスを用いれば、オブジェクトの移動と回転を同時に指示できる。

また、階層メニューからアイテムを選択する場面では、階層が深くなる毎に別のポインティングデバイスを用いて指示をする事ができるので、ポインティングデバイスを移動する手間を低減できる。

**CSCW** 一つの画面を、同じ場所にいる複数のユーザーで共有するような場合、各ユーザーは、物理的に共有された画面上の様々な箇所を独立にポイントできる。

#### 3.2 MPISの形態

はじめに、ポインティングデバイスの分類を示す。これらの違いはSPISにおいてはそれほど顕著な差ではなかったが、MPISにおいては、その特性を大きく左右するものとなる。

**物理デバイスの有無** ポインティングデバイスは、何らかの物体を手で操作するもの(例: マウス)と、指先で直接ポイントするもの(例: タッチパネル)の二つに分類できる。物理デバイスを用いるものでは、ユーザーはデバイスから手を離してもデバイスは操作平面上で静止しており、その位置は保存される。対して指先でポイントする場合は、ユーザーが手を離すとその座標は保存されない。本論文では以降特に示さない限り、デバイスという単語を、マウス型の入力デバイスで使用する物理デバイスの意味で用いる。

**直接 / 間接入力** ディスプレイと操作平面が一致している形態を直接入力、一致しない場合を間接入力と呼ぶ。直接入力の場合、ユーザーは画面上を指先やデバイスを用いて直接ポイントする事ができる。間接入力の場合、ディスプレイには、指先またはデバイスが画面上のどこをポイントしているかを示すために、ポインターを表示する必要がある。

#### 3.3 MPISの分類

前節で示した分類に従うと、MPISは四種類の形態が考えられる。

**間接 - 有デバイス** ユーザーは操作平面上で複数のデバイスをを用いる。各デバイスの操作平面上での位置が、そのまま入力座標となり、デバイスの位置関係と画面上のポインターの位置関係は対応している。デバイスが複数ある事に起因する問題がある。ユーザーがあるポインターを操作する時、対応するデバイスにその時点で触れていなかった場合、ユーザーはまずそのデバイスに触れなければならない。デバイスが一つであれば、手探りでデバイスに触れる事も可能だが、複数ある場合は、ユーザーはデバイスが置かれている操作平面を目で見なければならない。そのための視線移動のコストが発生する。

デバイスは汎用なもの他に、操作対象に特化したものを使用する事もできる。その場合、デバイスの形状や色等を、操作対象との関連性の強いものを選択できる。

**間接 - 無デバイス** ユーザーは操作平面上に両手を置き、指先でポインターする。ポインターは、指先で操作平面に触れた時に画面上に表示される。この形態では、ユーザーは操作平面を目で見る必要はなく、視線移動のコストは発生しない。

**直接 - 有デバイス** ユーザーは画面上、または画面が投影された卓上にデバイスを配置し操作する。間接 - 有デバイスとの違いは、対応する画面上の操作対象が簡単に視認できるので特にデバイスの形状を操作対象に特化せずとも対応関係が簡単に把握できる事にある。ただし、デバイスの大きさや形状によっては、画面がデバイスに隠されてしまうため、操作対象の視認が困難になる。

**直接 - 無デバイス** ユーザーは画面上の操作対象を直接指先でポイントするため、間接 - 無デバイスに比べてより直感的な操作感が得られる。しかし、指先や手によって画面がさえぎられるため、視認性は低くなる。

### 3.4 デバイスと操作対象の関連付け

間接 - 有デバイスの MPIS において、画面上の操作対象に特化したデバイスを用いる場合、デバイスと操作対象との関連は固定される。一方、一般的な GUI 環境では入力デバイスと操作対象との関連は固定されていない。本論文では前者のようなデバイスを専用デバイス、後者を汎用デバイスと呼んで区別する。

MPIS において専用デバイスを用いる事の利点は、デバイスと操作対象との関連が一目で把握できるという点にある。操作平面上で使用するデバイスの数が増えるに従って、デバイスと画面上の操作対象との関連を把握するのは困難になる。

しかし、デバイスと操作対象の関連は固定されているので、使用する可能性のある操作対象の数だけデバイスを用意しなければならない。また、画面上の操作対象の種類や配置がシステム側から変更されたりする場合、例えば過去の配置を再現したりアプリケーションを切り替えたりする場合、操作平面上のデバイスが自動的に移動しない限り、ユーザーがデバイスをシステムが示す位置に配置し直す手間が生じる。

MPIS において汎用デバイスを用いる場合、専用デバイスの欠点として挙げた手間は生じないが、画面上の操作対象とデバイスとを関連付ける手間が生じ、またその関連をユーザーは把握している必要がある。SPIS ではその関連は唯一つであるため、左記の手間は問題にならないが、MPIS においてはその手間は使用するデバイスの数だけ生じるので、これらの手間を低減するための工夫が必要となる。

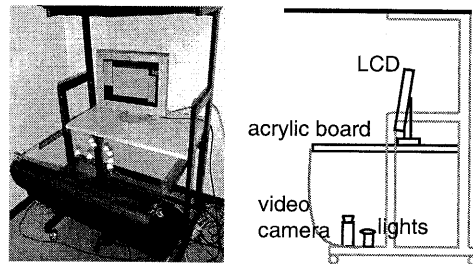


図1 プロトタイプシステム全体像

## 4. プロトタイプ実装

我々は MPIS のプロトタイプとして、実現するのが容易な、間接入力 - 有デバイス型の MPIS を構築した。このプロトタイプはビデオ入力を用いた独自のもので、同時に 8 箇所までポイントできる。

既存のポインティングデバイスによる MPIS の構築は検討の結果不適であると判断した。以下に簡単にその検討内容を示す。

USB マウスを複数接続して使うのは、各デバイスが大きく、指先で扱うのは難しい。また、絶対座標値を計測できないため、複数使用した際に各デバイスの座標系の不整合に起因する問題が生じる。例えば両手でマウスを同じ向きに並行に動かしたつもりでも、それぞれのマウスの向きが違えば画面上のポインターの動きは同じ向きにはならない。また、操作平面上ではマウス同士を接触させたつもりでも、画面上にそれが反映されない画面が生じる(図2)。

Wacom 社のタブレットは現在は同時に二個までしか入力デバイスを扱えない。ポヒマスセンサーは有線であるため、卓上で複数個扱うのには不向きである。

### 4.1 ハードウェア

透明なアクリル板<sup>\*</sup>を天板に用いた机を使用し、各デバイスはその上で操作する。その様子を板の下側からビデオカメラ<sup>\*\*</sup>で撮影し、その画像を解析して各デバイスの座標を計測する。図1に全体像を示す。

各デバイスは、底面にデバイス毎に異なる色の付いたフェルト生地を貼る。この色をもとに各デバイスを識別する。フェルト生地を用いた理由は、アクリル板の上で操作する上で摩擦を軽減するためである。デバイスの形状には制約はないが、ビデオカメラから見て余計な部分が見えないようにフェルト生地を貼る必要がある。底面の形には制約はないが、認識精度を向上させるため、ある程度の大きさ(現状では1cm角程度)が必要となる。デバイスにボタン等を付加して、位置情報以外の情報を加える事はできないが、後述する方法により、デバイスの

<sup>\*</sup> 5mm 厚のもので、少したわむが実験には支障はなかった

<sup>\*\*</sup> SONY DCR TRV900

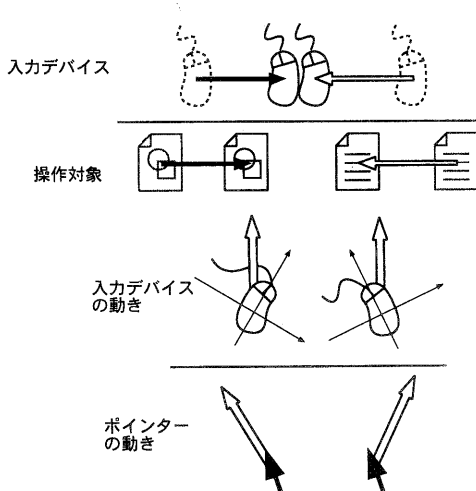


図2 間接座標入力に起因する問題

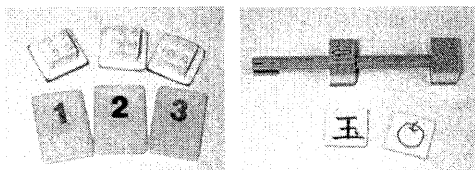


図3 左は汎用デバイスの例。右は専用デバイスの例で、上のものは動きに物理的制約のあるもの

高さまたは傾きを計測する事は可能である。図3に示すように、デバイスは汎用デバイスでも専用デバイスでも良い。汎用デバイスには主に16mm角のLEGOブロックを用いている。

画面は、机の上に置いた液晶ディスプレイに表示する。直接入力を実現するために液晶プロジェクターを使用して投影する事も検討したが、投影するためには板は半透明なものを使用する必要がある。しかしその場合、板の下から撮影した画像は彩度が不足し、計測に支障をきたす事がわかっている。また、投影された画像が誤認識の原因ともなる。これは偏光板を使う事で解決できるが、その場合入力画像の明度が減少し、計測に支障をきたす。

ビデオカメラは高速シャッターモード(1/1000秒)に設定している。これは、移動するデバイスを撮影したときに生じるブレを極力抑えるためである。底面の像がブレると背景と混色されてしまい、色を正しく識別できなくなる。

入力画像のコントラストを向上させるため、机の下部に光源を設置し、デバイスの底面を照明している。色に影響を与えないよう、白色に近い光源を用いた。また、ビデオカメラを高速シャッターモードに設定しているため、蛍光灯等の点滅する光源は使用できない。また、外

光の影響を抑えるために、各所に遮光板を設けている。

#### 4.2 ビデオ画像解析

ビデオカメラはPCに接続されており、このPCで画像を解析し、各デバイスの座標を計測している<sup>\*</sup>。図4に画像解析の様子を示す。

ビデオカメラからの入力画像はキャプチャーカード<sup>\*\*</sup>により、320×240ピクセルで約50000色の解像度のデジタル画像に変換される(図4-1)。この画像データから、あらかじめ登録された各デバイスの底面の色を抽出する(図4-2)。このとき、CCDの特性のために入力値にばらつきが生じると、机上の位置によって照明の強さに差があるため、同じ底面の色でもその入力値には幅が生じる。そこで、登録する各デバイスの底面の色は、あらかじめ幅を持たせて登録しておく。

次に、ノイズを除去する(図4-3)。ノイズ除去アルゴリズムは、注目しているピクセルおよび近傍8ピクセルのうち、8ピクセル以上のピクセルが同じデバイスの色であればそれを採用し、そうでなければノイズとみなし除去する、というものである。

こうして残ったピクセルの座標の平均値を、対応するデバイスの座標値として用いる。この座標値を、640×480ピクセルに拡大する。デバイスの底面が十分に大きくてノイズが少なければ、拡大しても充分な精度が得られる。経験的には1cm各程度あればよい(ノイズ除去後で400ピクセル程度)。

以上の処理は非常に簡単であるが故に、高速な計測が可能となっている。現行の実装では毎秒30scanを達成しており、これはビデオ入力の上限(毎秒30フレーム)から来る限界に達している。

加えて今回構築したシステムでは、デバイスに対応するピクセルの数を入力の一つとして計測している。あるデバイスのピクセル数が、デバイスがアクリル板に接地している時のピクセル数よりも少なければ、そのデバイスは持ち上げられているか傾けられていると判断できる。本システムではこの情報を、デバイスに対するユーザーの操作のひとつとして採用した。ただし、持ち上げ操作と傾け操作を識別する事はできない。現行の実装では、接地時のピクセル数の90%程度に閾値を設けると、操作平面のどの箇所においてもこの操作を誤らずに認識する事ができる。これは、アクリル板から2cm程度の高さに持ち上げる事に相当する。理想的には、操作平面とデバイス底面とが接触しているかどうかを認識するべきだが、今回の実装では認識する事はできない。

#### 4.3 仕様

本プロトタイプの仕様を表1に示す。

#### 4.4 汎用デバイス使用時の関連付け操作

3.4節で述べたように、汎用デバイスを用いる場合、

<sup>\*</sup> PCはAMD Athlon600 プロセッサを搭載したATX仕様のもので、OSにはLinux-2.2.14 kernelを用いた。

<sup>\*\*</sup> Bt848A チップを使用

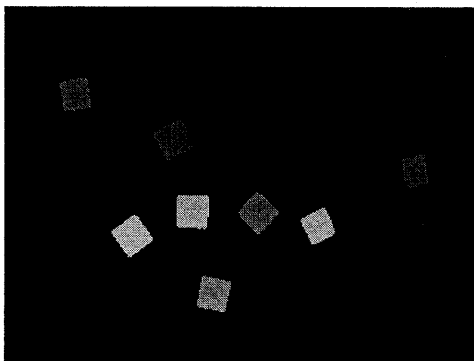
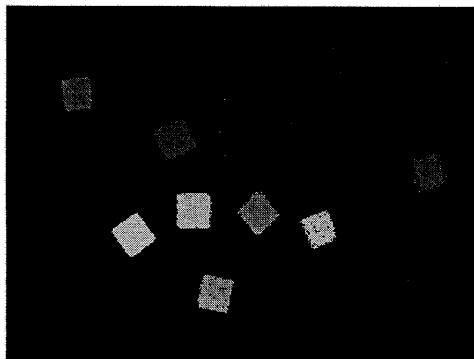
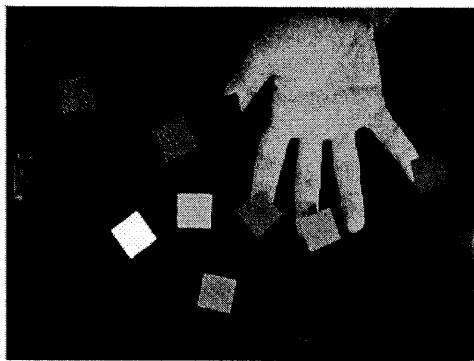


図4 画像解析の各段階

画面上の操作対象とデバイスとの関連付け操作を実装する必要がある。一般的なポインティングデバイスではボタンを用いて、1) ポインターを操作対象に重ねて、2) ボタンを押下する、という手順で関連付けを指示するが、今回作成した汎用デバイスではボタンのような入力装置は付加できない。そこで、今回作成した MPIS では以下の二種類の手法を実装した。

デバイスの持ち上げ操作による指示 デバイスを持ち上げると、画面上の対応するポインターを半透明表示される。この状態でポインターを対象とするオブジェクトに重ね、デバイスを操作平面におろすと、

表1 プロトタイプシステム仕様

|          |             |
|----------|-------------|
| 同時入力ポイント | 8箇所         |
| 計測速度     | 30回/秒       |
| 座標値解像度   | 640×480ピクセル |
| 操作平面実寸   | 320mm×240mm |

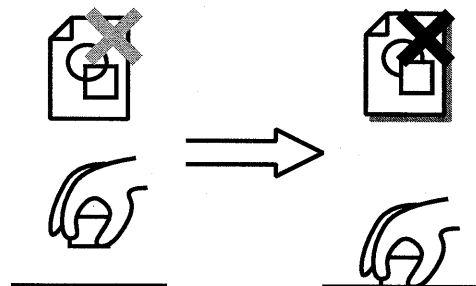


図5 デバイスの持ち上げ操作による関連付け操作

デバイスと操作対象が関連付けられる(図5)。またデバイスを持ち上げると、この関連は切り離される。

現在の実装では、持ち上げ操作を認識できる高さの閾値は約2cmである。デバイスを持ち上げるコストが高いのと、デバイスを持っている手では他のデバイスを操作できないために並列度が減じるという欠点がある。

ポインターの接触による指示 ポインターをオブジェクトに接触させると、デバイスと操作対象が関連付けられる。前述のコストは生じないが、目的のオブジェクトとポインターとの間に別のオブジェクトがある場合等に、ポインターの動きが制限されてしまう。また、関連付けの切り離しを指示する方法がないので、デバイスの持ち上げ操作を併用するか、画面上の特定の場所にオブジェクトを持っていくと切り離す、といった処理が必要となる。

## 5. アプリケーション

前章のプロトタイプシステム上で動作する評価用アプリケーションを作成した。

### 5.1 スライダー

図6は、スライダーを複数配置したアプリケーションで、オーディオミキサーの操作卓を模している。それぞれのスライダーには一つデバイスが対応しており、デバイスの操作平面上での奥行き方向の座標値がスライダーの値に対応している。左右方向の座標値は使用しない。現在の実装では、デバイスとスライダーの対応は固定されているため、デバイスを入れ換えると、画面上のスライダーの位置関係と、操作平面上のデバイスの位置関係は対応しなくなる。また、各デバイスの移動には物理的制約を設けていない。

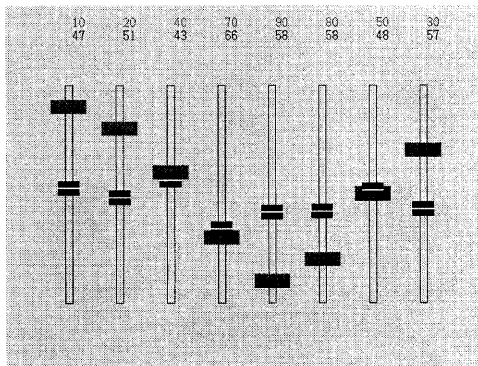


図6 アプリケーション: スライダー

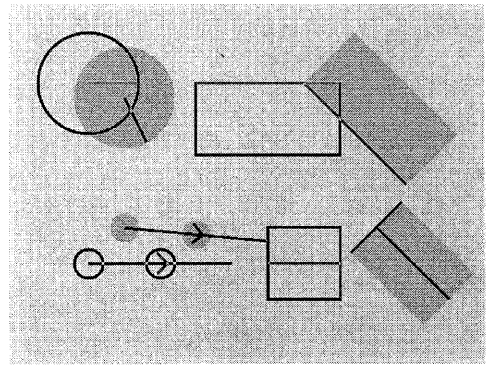


図8 アプリケーション: 位置合わせ (専用デバイス)

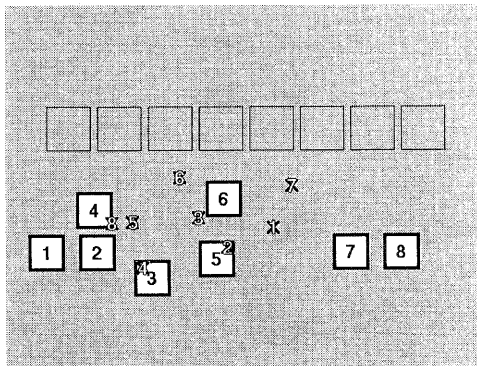


図7 アプリケーション: ソート (汎用デバイス)

## 5.2 ソート

画面上には数字が書かれた8個の矩形が置かれており、被験者はそれらを数字の小さい順に、決められた箇所に整列させる事が求められる。図7の下部に配置されているのが操作対象となる矩形である。画面上部にはこれらの札を配置させる場所を枠で示している。このアプリケーションでは専用デバイスと汎用デバイスの両方を実装している。汎用デバイスを用いる場合の関連付け操作は、4.4節で述べた二つの方法をそれぞれ実装した。ポインターと操作対象との接触を関連付け操作とする方法では、札が目的の矩形の中に入った時に関連付けを自動的に解消するようにしている。

## 5.3 位置合わせ

被験者は四つのオブジェクトを操作する。画面上には各オブジェクト毎に目的の場所と向きが示されており、被験者はすべてのオブジェクトを、指定された位置と向きに合わせる事が求められる。四つのオブジェクトのうち二つは、さらに長さが可変であり、指定された長さに合わせる事が求められる。

このアプリケーションでは専用デバイスと汎用デバイスの両方を実装している。専用デバイスを用いる場合

は、これらのオブジェクトと同等の形をした入力デバイスを提供する。長さが可変なオブジェクトでは、入力デバイスも、物理的に長さが可変な入力デバイスを提供する。

汎用デバイスを用いる場合は、以下の手順でオブジェクトを操作する。まず、オブジェクトに最初に関連付けした入力デバイスで、オブジェクトの位置を動かす事ができる。そして二番目に関連付けた入力デバイスで、オブジェクトの向きや長さを操作できる。

## 5.4 観察結果

現時点ではまだ評価実験は実施していない。本節では、2名の被験者に上記のアプリケーションを操作してもらった時の観察から得た知見を述べる。

スライダーの操作から、複数のデバイスによる並行操作は効果がある事が分かった。ユーザーには、スライダーを合わせる先の目標を、ある程度の幅を与えて課している。この幅が大きい場合、すなわち値合わせに精度が要求されない場合は並行操作の効果が表れる。しかし、幅を小さくして精度が要求されるようになると、ユーザーは微調整の為に各デバイスを逐次的に動かすようになった。

複数のデバイスを用いる事による並行操作の効果はソート操作にも見られるが、微調整段階での並列度の低下がやはり見られた。

ソート操作では、関連付け操作を、持ち上げ操作からポインターの接触操作に変える事で、課題全体の達成時間が短くなる傾向が見られた。特に、関連付けの為にデバイスを持ち上げている間は並行操作がまったく行われない傾向がある。これは、現在の実装ではデバイスを持ち上げている間は得られる座標値が不安定になり、ユーザーが意図した場所をポイントしたままデバイスを降ろすのが困難な事も一因である。一方、接触操作によ

※ 番号付けされた矩形の大きさは50ピクセル四方で、目標位置を示す枠は60ピクセル四方であり、10ピクセル程度の遊びでは並行操作の効果は得られないと考えられる。

る関連付けの場合、操作しているデバイスに、ユーザーが意図していなかった操作対象が関連付けられる事が度々あったのは予想の範囲内だが、その場合すぐに別のデバイスを持ってきてそのデバイスに本来目的としていた操作対象を関連付けるという操作が見られた。

位置合わせのアプリケーションでは、やはり関連付けのコストが無い、専用デバイスを用いる環境の方が達成時間が短い。ただし個々のデバイスが大きいためか、せいぜい二並列の並行操作しか見られなかった。

すべてのアプリケーションに共通する結果として、操作平面と表示画面との間での視線移動が頻繁に発生している。被験者への聞き取り調査では、操作しようとしているデバイスの場所を確認するために、表示画面から操作平面へと視線を落す必要があった事がわかった。また、視線は表示画面に向けた状態のまま、操作平面の上で手探りでデバイスを探す動作が何度か見られた。

## 6. 議論

両手による並行操作の有効性はすでに多くの研究で示されており<sup>1)5)3)</sup>、今回の観察結果にも並行操作の効果は見られる。指先の使用を前提とした並列度の高い並行操作についても、条件が整えば十分な効果があるものと予測する。

微調整段階で並列度が低下するのを防ぐには、システム側からの何らかの支援を検討する必要がある。例えばソート操作アプリケーションにおいては、各矩形を配置する先は既知であるので、矩形が配置先の枠に近付いたら自動的に枠内に移動させるという支援が考えられる。しかしスライダー操作のように、実アプリケーションにおいてはその目標値をシステム側が知る事ができないと考えられるような場面において有効な支援策は考えにくい。ただし実アプリケーションでは、今回実装したように目標値に合わせてスライダーを操作するのではなく、ユーザーはフィードバックを得ながら微調整を繰り返すのが一般的であり、今回実装した評価アプリケーションはそうした実アプリケーションの実態を反映していない。

デバイスと操作対象との関連付け操作に起因する並行操作の阻害は、間接 - 有デバイスの MPIS 個有の問題である。無デバイスの MPIS であれば単に指先で触れるだけで関連付け操作とすることが出来る。一方、有デバイスの MPIS では一度関連付けしたデバイスはその関連を保持し続けるという特性がある。例えば関連付けは逐次的に行い、その後には並行操作をする、というような事が可能である。

表示画面と操作平面との間での視線移動を抑制するには、一つには直接入力用の MPIS を用いれば良い。しかし今回の実験で、デバイスを探すための視線移動があったことから、間接 - 無デバイスの MPIS でもこうした視線移動を抑制できると推測する。

## 7. 今後の課題

十分な数の被験者による評価実験を実施する。評価用アプリケーションの種類を増し、汎用デバイスと専用デバイスの比較を重点的に行う予定である。

また、今回の実験で有デバイスの MPIS の問題点が明らかになったので、今後は無デバイスの MPIS を実装し、比較実験を行う。現在、間接 - 無デバイスの MPIS の検討をしており、今回実装したプロトタイプとその特性を比較する。

## 参考文献

- 1) Buxton, W. and Myers, B.: A study in two-handed input, *Proceedings of CHI'86*, pp. 321-326 (1986).
- 2) Fitzmaurice, G. and Buxton, W.: An Empirical Evaluation of Graspable User Interfaces: towards specialized, space-multiplexed input, *Proceedings of CHI'97*, pp. 43-50 (1997).
- 3) Hinckley, K., Pausch, Randy Proffitt, D., Pat-ten, J. and Kassell, N.: Cooperative Bimanual Action, *Proceedings of CHI'97*, pp. 27-34 (1997).
- 4) Ishii, H. and Ullmer, B.: Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms, *Proceedings of CHI'97*, pp. 234-241 (1997).
- 5) Kurtenbach, G., Fitzmaurice, G., Baudel, T. and Buxton, B.: The Design of a GUI Paradigm based on Tablets, Two-hands, and Transparency, *Proceedings of CHI'97*, pp. 35-42 (1997).
- 6) Myers, B. A., Stiel, H. and Gargiulo, R.: Collaboration Using Multiple PDAs Connected to a PC, *Proceedings of CSCW 98*, pp. 285-294 (1998).
- 7) Underkoffler, J. and Ishii, H.: Illuminating Light: An Optical Design Tool with a Luminous-Tangible Interface, *Proceedings of CHI'98*, pp. 542-549 (1997).
- 8) Underkoffler, J., Ullmer, B. and Ishii, H.: Emancipated Pixels: Real-World Graphics In The Luminous Room, *Proceedings of SIG-GRAPH'99* (1999).
- 9) 松下伸行, 綾塚祐二, 暦本純一: Dual Touch: ペン型 PDA のための新しい操作手法, インタラクティブシステムとソフトウェア VII: 日本ソフトウェア科学会 WISS'99 (安村通晃 (編)), 近代科学社, pp. 23-32 (1999).
- 10) 中村聡史, 塚本昌彦, 西尾章治郎: 2つのマウスを用いたウィンドウ操作機構の設計と実装, 情報処理学会研究報告, Vol. 99, No. 35, pp. 1-6 (1999).