ハンドジェスチャを用いた公共大画面向けインタフェース

中村 卓, 高橋 伸, 田中 二郎

筑波大学大学院システム情報工学研究科コンピュータサイエンス専攻

Hand Gesture Interaction for Large Public Display

Takashi Nakamura, Shin Takahashi, Jiro Tanaka

Graduate School of System and Information Engineering,University of Tsukuba

1 はじめに

近年，プラズマディスプレイやプロジェクタなどを利用した大画面が駅などの公共の場に広く普及してきた．しかし，その大画面とインタラクションを行うためのインタフェースは，マウスなどといった従来のインタフェースをそのまま利用しているのが現状である．そのようなインタフェースは大画面環境には不向きであり，また公共の場では設置場所などの問題がある．そこで，公共の場にある大画面を操作する方法として，ユーザのハンドジェスチャを利用したインタフェースを試作した．

本論文では，ハンドジェスチャを用いて大画面とのインタラクションを可能にする大画面向けインタフェースを試作した．これにより，様々な環境において，大画面とのインタラクションが容易になる．また，それに伴い，様々な場所で大画面をより広く利用することができるようになると考えられる．

2 インタフェースの現状

現在，駅の構内や街角など様々な場所に大画面が設置されている．このような大画面上では様々な情報が表示され，それを見ることができるが，見る側からは操作できないのがほとんどである．その原因のひとつとしてインタフェースの問題があると考えられる．

2.1 従来の公共端末のインタフェース

公共の場において，従来のインタフェースを利用するのは非常に困難である．例えば，マウスを利用したインタフェースの場合，マウスを利用するための水平なスペースを確保するのは大変である．タッチパネルを利用したインタフェースの場合は，手が届かない場所での操作が困難であるため，自然と画面のサイズが限定されてしまう．また，マウスやペンなどの従来のインタフェースは，本来は PDA などの小さなディスプレイやコンピュータのディスプレイなど小中規模のディスプレイ向けのインタフェースであるため，壁サイズのディスプレイなどといった大規模なディスプレイ向けのインタフェースではない．そのため，従来のインタフェースをそのまま利用するのは適切ではない．

2.2 大画面向けインタフェース

大画面向けのインタフェースの研究は盛んに行われてきている．大画面を操作するという試みは四半世紀以上前からすでに行われてきている [7]．また，最近では，手にデバイスを装着して大画面を指差すことでカーソルを動かしたり，指を動かすことでクリックの動作を行うことができるインタフェースがある [4]．一方，手ではなく，携帯電話からの色相差を利用した光信号を送ることでカーソルを動かしたり画像を貼り付けたりすることができるインタフェースの研究も存在する [5]．

しかし，このようなインタフェースのほとんどは手に何かしらのデバイスを装着したり，専用のソフトウェアをあらかじめ用意しなければならないといった問題点があるため，公共向けのインタフェースとはいえない．

そこで，本論文ではこのような問題を解決する方法として，ハンドジェスチャを用いたインタフェースを提案し，試作システム Hands-In を作成した．

3 Hands-In

公共大画面を操作するためのインタフェースとして，手を利用したインタフェース Hands-In を作成した．Hands-In の特徴は，両手によるハンドジェスチャを入力として用いたインタフェースである．このシステムでは，図 1 のようにユーザは大画面と対面した形で操作することができる．

Hands-In は，ユーザ側にはデバイス等は装着させない．また，両手への動作の割り当ては，手の運動パターンを基に割り当てた．

3.1 設計指針

手を用いたインタフェースでは，手にセンサのついた手袋などのデバイスを装着することが多い．それには様々な理由があるが，デバイスを用いる最大の理由はより正確でより細かい動作を行いたいためである．しかし，本研究は公共の大画面で利用可能なインタフェースの設計が目的である．公共の場におけるインタフェースはその場で誰でも使えるものが好ましい．そのため，本システムでは，手に何も持たない，何もつけないことを前提にしたジェスチャを作成し，インタフェースを設計する．

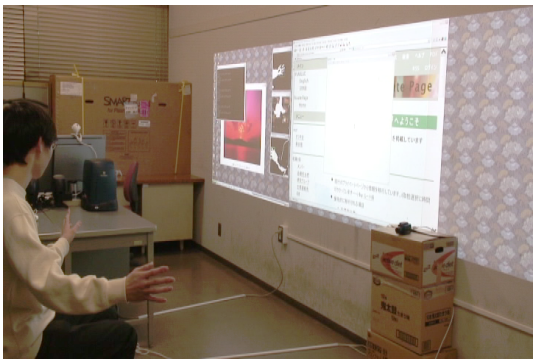

図 1: 利用イメージ

また，両手による動作の割り当て方は，本システムでは基本は右手でカーソルの移動を行い，左手には決定などの補助的な動作を割り当てることにした．これは，利き手とそうではない手の運動パターンを考慮している [3]．両手を同時に利用した動作についても，基本的に左手でその動作の開始や終了を行い，右手で範囲の選択を行う．これにより，よりわかりやすい動作になると思われる．

3.2 ハンドジェスチャによる画面の操作

本システムでは，手を開いているか閉じているかと，手がどの位置にあるかという情報からジェスチャを設計した．例えば，右手を開いた状態のまま動かしたら，その移動量に応じて相対位置座標指定によるカーソルの移動を行う．

本システムで定義した操作をまとめると表 1 のようになる．表 1 の open は手を開いている状態，grab は手を閉じた状態，hold は手を握った状態を維持することを示す．また，○や×はその動作が行われているかどうかを示し，-はその状態に依存しないことを示している．そして，upper・center・under は手がどの位置にあるかを示している．

表 1: 操作一覧

右手		左手		
形状	移動	形状	位置	動作
open	○	-	-	相対位置座標指定での移動
grab	○	-	-	絶対位置座標指定での移動
-	○	open→grab	center	マウスの左クリックダウン
-	○	hold	center	ドラッグ
-	-	hold→open	center	マウスの左クリックアップ
-	×	open→grab	not center	スクロールの開始
-	-	hold	upper	スクロールアップ
-	-	hold	under	スクロールダウン
-	-	hold→open	-	スクロールの終了
-	×	open→hold	center	Popie を起動

インタフェースを設計する上で，タッチパネルディスプレイのような絶対位置座標指定による移動か，マウスやジョイスティックのような相対位置座標指定による移動を利用するかは設計するシステムで異なってくる．本システムでは，絶対座標指定と相対座標指定の二つを使い分けることにした．絶対位置座標指定のみでは，細かい位置にカーソルを持っていくのが困難で，また相対位置座標指定では，すばやいカーソルの移動が難しいためである．大体の位置にカーソルを持っていきたい場合には絶対位置座標を利用し，細かい移動を行う際には相対位置座標指定を利用した．

利き手の方がもう片方の手と比べて細かい動きが可能なため，カーソルの移動は全て利き手で行う．また，絶対位置座標指定による移動か相対位置座標指定による移動は右手の形状によって判断する．手を開いた状態で移動させた場合は，現在の手の位置とその直前の手の位置からカーソルの移動量と方向を決定する相対位置座標指定による移動を行う．また，手を閉じた状態であれば，キャプチャした画像中の手の位置をもとに絶対位置座標指定による移動を行う．

利き手でないほうの手では，利き手と比べて大まかな動作しかできないため，画面のスクロールなど大まかな動きで十分な動作を行う．また，ドラッグなどの両手を利用する動作については，利き手ではない手で動作の開始や終了を行うようにした．例えば，利き手でない手が中心付近で握られて，かつ利き手でカーソルの移動を行った場合，マウスでいうドラッグの動作が行われる．ドラッグ中でも，相対位置座標指定と絶対位置座標指定の 2 種類のカーソルの移動が行えるため，広い範囲でドラッグを行う場合は絶対位置座標指定で，狭い範囲の場合は相対位置座標指定と使い分けることができる．

3.3 公共大画面での利用例

本システムは，様々な公共の場にある大画面での利用を想定している．そこで，公共の場での利用例についても考えてみた．本論文では，駅に設置されている大画面と高速道路のサービスエリアなどに置かれている大画面の 2 つの例について考えてみる．

3.3.1 駅での利用例

例えば，画面上に乗り換え案内や時刻表などがウィンドウとして複数表示されている．大画面上での複数表示なので，小さなウィンドウでもある程度の情報を読み取ることは可能である．見たいにカーソルを持っていくことでそのウィンドウのみが拡大されて表示され，詳細情報の閲覧や経路探索，店などの検索を行うことができる．

このとき，ウィンドウの選択には絶対位置座標指定による移動で選択を行い，ウィンドウ内では相対位置座標指定によってカーソルを操作する．また，後述する文字入力方法を用いることで文字列による検索なども行うことができる．

3.3.2 高速道路での利用例

高速道路のサービスエリアでは，渋滞箇所や混雑箇所が色で表示される．また，事故などはその場所に印がつけられる．しかし，広範囲の情報を表示しているため，何 km などといった詳細な情報が表示されていない．そこで，詳細情報を知りたいところにカーソルを絶対位置座標指定によってカーソルを移動させて選択することでその部分を一部拡大表示し，その詳細な情報を表示する．また，このとき，画面全体を利用して拡大するのではなく，選択した部分の周囲を利用して拡大する．これにより，全体の情報を見ながら知りたい情報のみを詳しく知る

ことができるので，操作している以外のユーザも全体の情報を眺めることができる．

4 Hands-In の実装

第 3 節で述べた内容を元に Hands-In の試作システムを作成した．試作システムは図 1 のように，ユーザは大画面と向かい合った状態で操作する．そのため，カメラはユーザを撮るように設置する．大まかなシステムの流れは，図 2 のようになる．カメラはユーザの手の動きをキャプチャし，そのキャプチャした画像を基に計算機内部で処理し，処理結果を基に動作などを決定し，それを画面上に反映する．

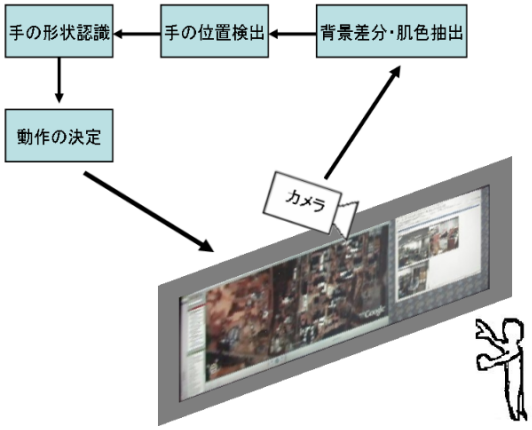

図 2: システム図

4.1 左右の手の領域分け

カメラからのキャプチャ画像を処理する際，試作システムでは，カメラ画像内で予め領域を定めた．左右の手の処理は，基本的にその領域内で全ての処理を行う．具体的には，カメラ 1 台のみで処理する場合には，図 3 のように右側の矩形領域を右手の領域として，左側の矩形領域を左手の領域として処理をする．なお，カメラのキャプチャ画像は予め鏡像反転させた画像を用いる．左右の手がそれぞれ領域内のどの位置にあるかを調べることで，手の位置を把握する．カメラを 2 台利用する場合には，1 台を右手用として，もう 1 台を左手用として割り当てればよい．

4.2 手の処理

カメラからキャプチャした画像について，まず，予め撮影した背景との差分をとることで，背景と人物を分離する．次に，その画像について色相差などを利用して肌色の部分だけを取り出す．最後に，ノイズを取り除いて二値化することで肌色の部分だけを抽出した．

肌色を抽出した後に，手がどの位置にあるかを調べる．右手の場合は，まず，抽出した手の画像の中で肌色が最も右側の点と

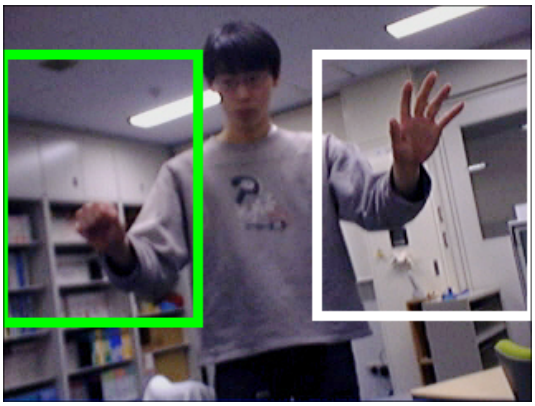

図 3: 領域の割り当て

最も上の点を調べる．次に，最も右にある点の x 座標と最も上にある点の y 座標を基に最も右上の点を特定する．そして，その点から一定範囲内にある領域を手の領域として認識する．左手の場合は，最も左上の点を調べること以外は右手の場合と同じである．

本システムの動作の多くは，手がどの位置にあるかということを把握する必要がある．そのため，ユーザ側に手がどの位置にあるかという情報を明示する必要がある．そこで，ユーザへのフィードバックとして図 4 のような画像を大画面上に表示した．これによって，ユーザはどの位置に手があるかを把握することができる．また，図 4 中の矩形領域が手の領域として表示される．これによって，ユーザはある程度自分が行おうとしている動作を把握することができる．

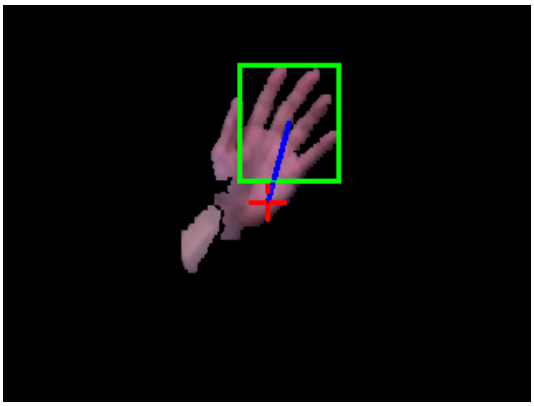

図 4: 手の処理の結果

4.3 動作の決定

動作を決定するに当たって，まずは，手の形状を調べる必要がある．手の形状の調べ方は様々な方法があるが，本システムでは，手が開いているか否かを指の数を調べることで判定している [6]．

指の数を調べるために，予め認識した手の画像領域について，次のような処理を施す．

1. 手の画像を何回か収縮処理を行う．
2. 1 の画像について，さらに膨張処理を何回か行う．

この方法は，指は手の平より細いことを利用している．このような処理を施すことで，収縮してから膨張処理を施した後の画像は，細い部分が削られた画像になる．そのため，処理前と処理後の画像を比較することで指の数を判断できる．それによって手の形状を判別している．

左右の手両方について位置と形状を判別した後に，その結果を元に動作を決定する．大抵の動作はそのときの両手の状態から決定・実行できるが，カーソルの移動など移動量と方向が必要な場合がある．例えば，絶対位置座標指定の場合は，その場合はキャプチャした画像の位置のみから決定できるが，相対位置座標指定の場合はそれだけでは方向や移動量を決めることができない．そのため，相対位置座標指定の場合は，そのときの結果とさらに一つ前の結果を元に移動する方向と移動量を決定する．

4.4 試用による評価

実際に作成した試作システムを実際に利用してみた．カメラからのキャプチャ画像を処理しているため，画像自体は秒間 5〜10 フレーム程度であったが，比較的スムーズに動作した．手の位置検出は 9 割程度の精度であったが，手の形状の認識については 7・8 割程度の精度であった．

5 本システムにおける文字入力

本システムでは，文字入力手法についても実装した．文字入力の手法として，佐藤ら [1] の Popie を改良した．Popie とは図 5 のような FlowMenu[2] を利用したペンベースのメニュインタフェースである．今回は Popie の機能の一つである文字入力機能のみを利用した．

Popie の文字入力機能の特徴は，子音を入力し，その入力に対する変換候補を選択することで文字入力を行う．例えば，“ 中村 ”と入力したい場合には，“ NKMR ”と入力して変換候補から“ 中村 ”を選択することで入力することができる．

しかし，Popie は元々ペン入力のためのインタフェースである．そのため，本研究では，手を用いたインタフェース向けに改良した．主な改良点として，子音入力部と変換候補選択部を分けることにした．具体的には，子音入力部を右手で，変換候補選択部を左手で行えるように改良した．本システムにおいて，文字入力の開始方法は，左手をキャプチャ画像の中心付近で一定時間手を握った状態を維持することで開始する，終了方法は，左手をキャプチャ画面に写らなくなったら終了する，また，文字入力中のカーソルの移動については，常に相対位置座標指定による移動を行うようにする．

6 まとめ

本論文では，公共大画面向けのインタフェースとして，両手のハンドジェスチャを利用したインタフェースとして Hands-In を提案，試作した．これによって，様々な公共の場にある大画面を利用しやすくなり，大画面の利用シーンが広がるのではないかと思われる．

図 5: FlowMenu の概観

今後は，より使いやすいインタフェースを目指すために，評価実験を行い，改良する．また，手を用いたインタフェースの場合，従来のようなポップアップメニュは使いにくいため，FlowMenu を利用したメニュを作成していく．具体的には，Popie を手を利用したインタフェース用に改良することが考えられる．そして，実際に公共の大画面で利用可能なアプリケーションについても設計試作していく．

参考文献

[1] 佐藤大介, 志築文太郎, 三浦元喜, 田中二郎．ペンの周囲で操作することを可能にするインタフェース． Proceedings of Workshop on Interactive Systems and software 2004 (WISS 2004), pp. 71-76.

[2] Francois Guimbretiere and Terry Winograd. FlowMenu: Combining command,text, and data entry. In *UIST2000*,pp. 213–216.

[3] Ken Hinckley, Randy Pausch, Dennis Proffitt,and Neal F. Kassell. Two-Handed Virtual Manipulation. In *ACM Transactions on Computer-Human Interaction, Vol. 5, No. 3*, September 1998, pp.260–302.

[4] Daniel Vogel and Ravin Balakrishnan. Distant Freehand Pointing and Clicking on Very Large. In *UIST'05*,pp.33–42,2005.

[5] Kento Miyaoku, Suguru Higashino, and Yoshinobu Tonomura. C-Blink: A Hue-Difference-Based Light Signal Marker for Large Screen Interaction via Any Mobile Terminal. In *UIST'04*,pp.147–156.

[6] 入江 耕太, 梅田 和昇, 若村 直弘．ジェスチャ認識を利用したマン・マシン・インタフェースの構築－インテリジェントルームへの適用－．日本機械学会 *[No.03-4]* ロボティクス・メカトロニクス講演会 *'03* 講演論文集,1A1-3F-D1(1)–1A1-3F-D1(2).

[7] Richard A. Bolt. “ Put-that-there ”: Voice and gesture at the graphics interface. *SIGGRAPH '80*,1980, pp.262–270.