

# ProphetCursor : 顔の向き方向によってカーソル移動を補助する 大画面ポインティング手法

程 浩侃<sup>\*1</sup> 高橋 伸<sup>\*1</sup> 田中 二郎<sup>\*1</sup>

## ProphetCursor: Large-screen Pointing Technique Using Cursor Movement Assisted by User's Face Orientation

Haokan Cheng<sup>\*1</sup>, Shin Takahashi<sup>\*1</sup> and Jiro Tanaka<sup>\*1</sup>

**Abstract** - We propose a novel method to perform operations on large-screen using cursor movement which is enhanced by user's face orientation, achieving both high speed and high accuracy in interactions. In the proposed method of our present study, detection of user's face has been implemented to estimate a set of on-screen objects on which the user is probably going to perform a selection operation. And within the filtered set of objects, users can use their hands to do more precise movement and selection. We describe in this paper the implementation of the method utilizing depth-based and conventional cameras, along with experiments evaluating the effectiveness of the technique.

**Keywords:** Large-screen interaction, pointing technique, face orientation, hand gesture

### 1. はじめに

LED や液晶技術などの各種のディスプレイ技術の発展により、大画面ディスプレイの設置コストが削減され、解像度などのパフォーマンスも向上されている。大画面ディスプレイを用いれば、大量な情報を一度に表示することが可能になる。また、タッチパネルや画像認識技術の発達により、ユーザが大型ディスプレイに対し能動的に操作を行うことが可能になり、この利点を利用した操作性を重視する大画面応用も盛んになっている。例えば、近年インタラクティブなデジタルサイネージの設置数が増加しており、操作を行うことでユーザは従来の紙の広告のように受動的に情報を取得するだけでなく、自ら欲しい情報を選んで閲覧することも可能になった。

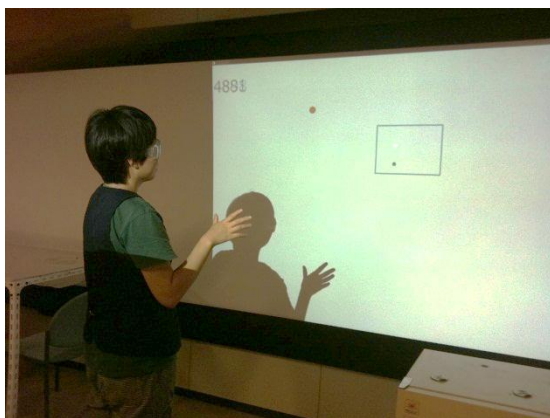


図 1 本研究提案手法の利用シーン

Fig.1 Utilization scene of the proposed technique

大画面を活用し、アイコン・画像・文字列などの操作対象を画面に大量に配置する利用シーンが多く、そういった場合でカーソルによるポインティング操作はディスプレイとの非接触性とピクセル選択による精密さを持つため、広い画面範囲内で正確な選択操作を行うための適切な方法だと考えられる。一方、PC モニターのような標準ディスプレイで用いられている従来のカーソル移動手法を大画面に適用する場合、カーソルの移動速度とポインティング精度の両立が困難であるという問題が存在する。具体的には、高解像度の画面においてマウスカーソルを長距離移動するためにカーソルの速度を増すと、細かい移動が難しくなり、精密なポインティング操作が困難になる。さらに、大画面ではカーソルがユーザの視野の外にある確率が高くなり、位置を特定しづらいという問題点も存在する。これらの問題は操作効率を低下させる恐れがある。

このような問題を解決するために、本研究では大画面操作環境において、画面全体に対するポインティング操作の効率を保ちながら局部への細かい操作を可能にする手法の実現を目的とする。この目的には「効率的な全体操作」と「精密な細部の操作」という二つの要件があると考えられる。前者を達成するためには、画面全域を把握するために画面からある程度離れた場所で操作を行う必要があり、その上で操作対象間での速やかな移動が必要だと考えられる。そして後者を実現するためには、ポインティング位置に対する調整機能が必要だと考えられる。

本稿では、第 2 章で本研究の提案手法について説明する。第 3 章では、距離カメラとレーザーポインタを用いるプロトタイプの実装について述べる。第 4 章でプロト

<sup>\*1</sup>: 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

<sup>\*1</sup>: Department of Computer Science, Graduate school of Systems and Information Engineering, University of Tsukuba

タイプシステムを実験環境とした評価実験について説明し、実験結果に対する考察を述べる。

## 2. 提案手法：ProphetCursor

### 2.1 提案手法の概要

本研究では、ProphetCursor と名付けた、ユーザの顔の向き方向と手の動きとの両方の情報を同時に利用する大画面ポインティング手法を提案する。この提案手法では、ユーザの画面に対する注目行為を利用し、従来のカーソル移動手法の補助を図る。図2で示すように、ユーザはまず、デスクトップアイコンやボタンなどの操作対象の大まかな位置に顔を向け、その位置の特定を試みる。そして手の動きでカーソルを調整し、正確な位置に移動させる。顔の向きを利用することにより、ユーザは画面内の任意の位置に注目点を置き、その位置情報を利用して操作対象の絞り込みを行える。そして手の動きで限定された対象に対し、操作を決定することができる。この二段階の動きにより、速やかな移動操作と精密な選択操作を図っている。

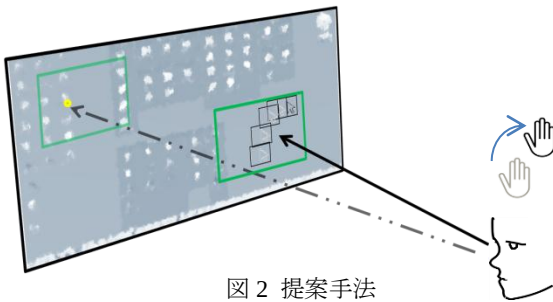


図2 提案手法  
Fig.2 The proposed technique

### 2.2 注目位置情報を用いる操作の利弊

注目位置情報を用いる利点の一つとして、ある対象を「見る」という行為は人の体では常に行われている行為であり、非常に直感的だと考えられる。そして本手法にとって特に重要な利点として、視線の移動速度が速いため、画面全体を簡単に見渡すことができ、操作対象の周辺位置を速やかに特定できる、という点が挙げられる。

一方、人の視線は常に小刻みに変化していて、確定的な注目点は取得しづらいという問題が存在し、多くの場合では測定位置の平均値を使用しているため、測定値と実際の注目点位置にはズレがあり、精度に問題がある。そのため、視線によるポインティングは細かい操作には向かないと考えられる。また、視線を直接操作に使用するとき、注目先とカーソル位置の不一致や、視線で常時ポインティングするときに意図しない操作を行ってしまうという Midas Touch 問題[1]など、操作の違和感を招く要因が存在し、結果的には操作効率に悪影響を与えてしまうと考えられる。

本手法では、人の視覚認知が視野の範囲で行われるという特徴に着目し、視線で直接対象を操作するのではな

く、大まかな注目位置を中心とした平面領域を利用して、注目行為通常の操作の補助として用いる。視線測定に技術的な難点が存在するという現状の中で、この手法を通して、注目位置情報を利用して操作効率と精度の向上を図る。

### 2.3 カーソル移動

2.1 提案手法の概要で述べたように、本手法で提案するカーソル移動は顔の向きによる操作範囲の絞り込みと手の動きによる細かい移動の二段階に分けられる。以下にこの二段階の具体的なデザインを述べる。

#### 2.3.1 顔の向きによる操作範囲の絞り込み

画面を注目するときのユーザの顔の向きから、注目点の大まかな位置を取得する。大画面の所在平面でこの注目点を中心とした矩形の範囲を利用し、最終的に選択操作を確定するためにまず目標の操作対象の周辺をこの矩形が囲む範囲で絞り込む。この矩形範囲は注目点が取得されたときに生成され、常に注目点に従って移動し、注目点が画面から外されるときに削除される。

矩形範囲が存在するとき、カーソルの移動範囲がこの範囲に限定される。視覚フィードバックとして、画面に矩形の枠を描画し、カーソルの移動可能な範囲としてユーザに提示する。

この注目点には大まかな位置のみが求められているため、視線測定の精度問題が回避される。また、顔の向きを使用する為、ユーザは常に操作対象またはカーソルに注目する必要がなく、自然な操作感覚が得られると考えられる。

#### 2.3.2 手の動きによる細部移動

注目位置で決められた矩形の範囲内で、ユーザは手の動きでカーソルの移動を制御する。カーソルの移動速度は手の実世界座標における移動で決められる。また、図3のように、カーソルが範囲の枠に当たるとき、その枠の移動方向に押されて移動するようになっている。これにより、カーソルは常に矩形の「移動可能範囲」に入れてカーソルを持ち運ぶことができる。

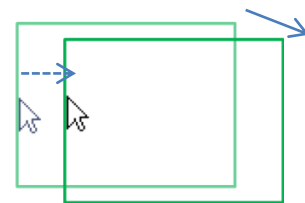


図3 矩形範囲とカーソルの動き  
Fig.3 Movement of cursor and rectangular

このように、ユーザの顔の向き方向を用いることで、カーソルの速やかな移動が可能となり、画面全体の把握と操作対象の速やかな切替えができる。そして手の動きでカーソルの位置を調整することにより、長距離移動後の精密操作の問題を解決できる。カーソルが常に視線の近くにあるため、カーソル自体の位置も特定しやすくな

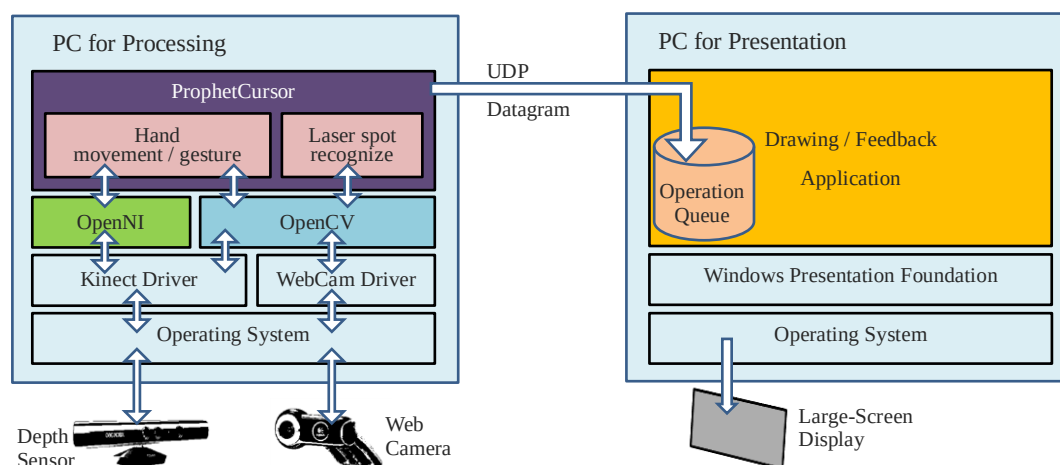


図 4 ハードウェアとソフトウェア構成  
Fig.4 Hardware and software configuration

り、効率と精密性が同時に向上されることが考えられる。

### 3. プロトタイプの実装

提案手法の有効性を確認するために、プロトタイプシステムを作成した。ハードウェアとソフトウェアの構成を図 4 に、システムの概観を図 5 に示す。

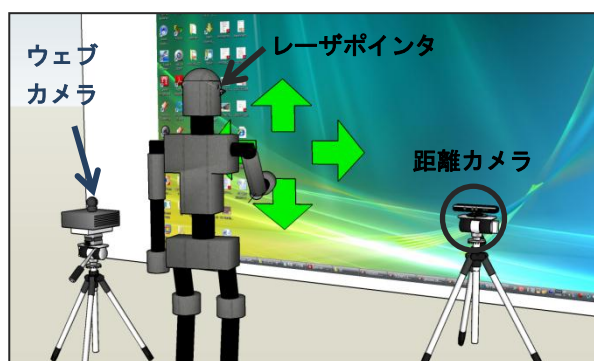


図 5 プロトタイプの概観  
Fig.5 Overview of the prototype system

図 5 で示すように、このプロトタイプでは、ユーザと大画面の間に距離カメラを設置し、画面前方の空間にある物体の距離情報により物体の空間位置を取得する。また、画面全体を撮影できる位置にウェブカメラを設置し、ユーザの頭部には前方向きのレーザーポインタを固定する。ウェブカメラにより、レーザーポインタが投射したレーザースポットの位置を特定する。そしてウェブカメラと距離カメラは 1 台の PC に接続され、同 PC が画像認識等の処理を行う。

また、操作の認識処理を行う PC とは別に、操作対象となるもう一台の PC を用いた。ユーザへのフィードバック機能・実験用ユーザインタフェースはすべてこの PC に実装している。二台の PC の間で UDP 通信を行い、処理用 PC から操作対象 PC へ処理結果を送るようソフトウェアの実装を行った。

ウェブカメラは Logicool Webcam Pro 9000 QCAM-200SX を使用し、 $640 \times 480 \times 24\text{bit}$  の画像を秒間

30 フレームのスピードでキャプチャする。距離カメラは Microsoft Kinect センサーを使用し、16bit のデータ（うち 11bit が距離情報）を  $640 \times 480$  の解像度、秒間 30 フレームのスピードで取得できる。

#### 3.1 顔の向き方向認識の実装

本実装ではユーザの頭部に装着されたレーザーポインタにより顔の向き方向を認識する。実装に使われたレーザーポインタが頭の向きに準じてレーザーの投射方向を変えるため、画面に投射されたレーザースポットの位置により、ユーザの大まかな注目点位置を測定できる。図 6 で示すように、レーザーポインタを安全メガネに固定し、ユーザに着用させる。そしてウェブカメラはレーザースポットを含めた大画面の映像を PC に送り、システム起動時に取得した（または手動で更新した）背景画像との差分によりレーザースポットの大画面における位置を得られる。レーザースポットの画面における相対位置を確定するために、システム設置時に一度スクリーンの四隅のカメラ映像における位置を指定し、この四点で得られた変換行列により射影変換を行い、レーザースポットのカメラ位置を画面の形に合わせて相対位置に変換する。



図 6 プロトタイプで使った  
レーザーポインタ（左）と距離カメラ（右）  
Fig.6 Laser pointer (left) and depth-based  
camera (right) used in the prototype

#### 3.2 手による操作の実装

##### 3.2.1 カーソル移動

手の動きの認識は距離カメラで行われている。本シス



テムでは距離カメラ（図 6）を用いて画面前方の物体の距離情報を取得し、その中から人体の各部位の位置情報を抽出する。この実装では Microsoft Kinect センサーを距離カメラとして用いて、人体部位の認識に OpenNI SDK を使用している。手の空間位置情報を利用し、画面と並行する二つの次元（X・Y 軸で形成された平面座標系における座標）の数値変化でカーソルの移動方向と移動速度の大きさを決める。通常のマウスでは長距離移動をするために、「移動→マウスを持ち上げ→マウスを卓上に置く→移動」のように複数回短距離移動を行うという方法が使用されているが、この実装でも同じ複数回の短距離移動を用いており、それを実現するために、ユーザの腕の伸ばし具合を利用した。具体的に言うと、頭・首・利き手側の肩の位置で確定される平面を基準とし、利き手からこの基準平面までの垂直距離を計算する。この距離がある閾値を越えたときにマウスの「机に置く」操作と見なし、カーソル移動を始め、閾値より低くなるときにマウスの「持ち上げ」操作として見なし、カーソル移動を停止させる（図 7）。

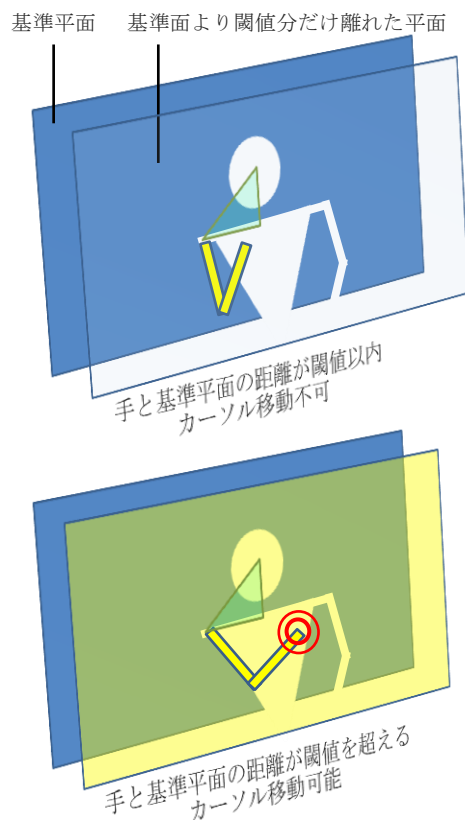


図 7 カーソル移動の開始と停止  
Fig.7 Stopping and starting cursor movement

### 3.2.2 選択操作

カーソル移動の他、操作対象に対し選択操作を行うために、手の「掴み」ジェスチャを実装した。即ち、手を握っている状態と指を伸ばしている状態にそれぞれ対象の選択・選択解除の機能を与える。このジェスチャを認

識するために、手の輪郭を用いた。カーソルの選択操作状態は以下の式（1）により決められる。

$$R_{shape} = \frac{|P_{contour}|^2}{|P_{shape}|} \quad (1)$$

$P_{contour}$  が手の輪郭のピクセルの集合を示し、 $P_{shape}$  が手のシルエットに含まれるピクセルの集合を示す。実装されたシステムは常に 5 フレーム分の  $R_{shape}$  値を記録し、連続な値の減少を検出すると選択操作を実行し、連続な値の増加を検出すると選択を解除する。

### 3.3 ソフトウェア構成

ソフトウェア構成としては、背景差分と手の形状認識のために OpenCV 2.1 SDK を使用し、距離情報の取得と人体各部位の位置特定には OpenNI 1.0.0 SDK を使用し、C++/CLI で処理機能を実装した。ウェブカメラのパラメータ調整には Logicool Webcam Software を使用した。

## 4. 実験と考察

提案手法の操作効率向上に対する有効性を調査するために定量的な面と定性的な面両方から評価実験を行った。それぞれの内容と結果、その考察について述べる。

### 4.1 実験の内容

被験者 6 名に所定の画面のランダムな位置に円形のターゲットを提示し、カーソルをターゲットの所在位置に動かして選択するというタスクを行ってもらった。（図 8）今回の実験では、移動時間を正確に取得するため、選択操作の実行方法として実行時間が不確定である「掴み」ジェスチャを使用せず、実行時間が一定となるように「停滞選択」を使用した。図 9 で示すように、カーソルをターゲットの円形範囲（図 9、ターゲット中心の丸型部分）に 700ms 停滞させることで選択操作を実行する。ターゲットの円形範囲から 700ms 経過せずに出てしまった場合は失敗した動作とし、その所要時間が実際の所有時間から減算される。カーソル移動には 3.2.1 に述べた複数回の短距離移動方法を実装する。被験者がタスクを行う間、カーソルの移動から選択操作が完了するまでの時間とカーソルの短距離移動回数を記録した。

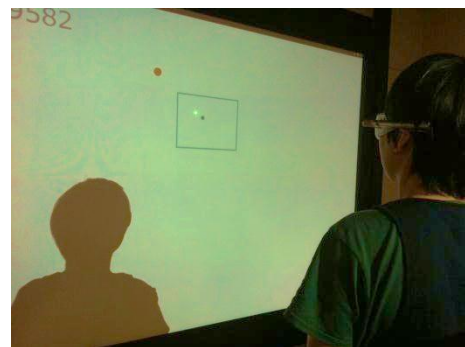


図 8 評価実験

Fig.8 Evaluation experiment in progress

本実験では以下二種類のタスクを行った：

- タスク 1：従来手法（手によるカーソル移動のみを使用）
- タスク 2：提案手法（顔向きと手によるカーソル移動両方を使用）

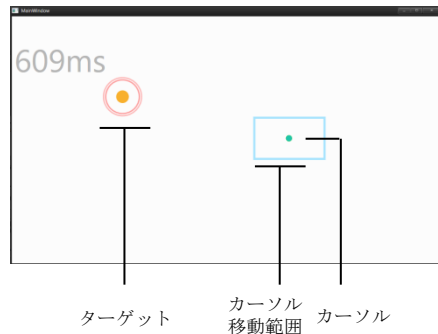


図 9 実験中の画面表示

Fig.9 On-screen contents in experiment

実験の手順として、まず被験者に 2 つの手法についての説明を行う。次に、それぞれの手法で操作の練習をしてもらう。その後、カーソルの移動速度を被験者の好みの速度に調整し、この速度設定で二つの手法を用いて各タスク 12 回ずつ行ってもらった。

実験環境として、横幅 200cm、高さ 150cm のプロジェクタ画面を使用し、解像度は 1600 ピクセル×1200 ピクセルとした。被験者は大画面から 200cm ほど離れたところでタスクを行った。腕の伸び具合に用いられる閾値を 300mm に設定した。また、連続で現れる二つのターゲットの距離がスクリーンの横幅の半分（800 ピクセル）とした。

#### 4.2 実験結果

各被験者のカーソル平均移動時間を図 10 に示す。エラーバーは各標準偏差を示している。6 人の被験者において、従来手法を用いたタスクでは、各被験者の平均所要時間は最短 4.2 秒、最長 12.2 秒で、平均 7.5 秒であった。一方、提案手法を用いたタスクでの所要時間は最短 3.9 秒、最長 7.1 秒、平均 5.1 秒となり、全ての被験者に於いて時間の短縮が示されていた。提案手法を用いたタスク 2 は従来手法を用いたタスク 1 に比べて、最短 0.3 秒、最大 4.9 秒の平均移動時間の短縮がみられる。

カーソルに対し短距離移動操作を行った回数を図 11 に示す。従来手法の平均移動回数は 4.07 回であり、提案手法は平均 2.31 回を示していた。移動回数の分布状況について、従来手法の移動回数は主に 2 回以上に集中し、被験者 6 名が行った全 72 回のターゲット選択タスクの内 2 度カーソル移動を行った回数は 27 回で、全体の 37.5% を占めていた。1 度のみ移動した回数は 2 回であり、2.78% の割合しか示していなかった。一方、提案手法では一度のみの移動回数は 29 回で全体の 40.3% を占めし、2 回以上移動した回数は従来手法と比べて全体的に少なくなったという結果が見られる。

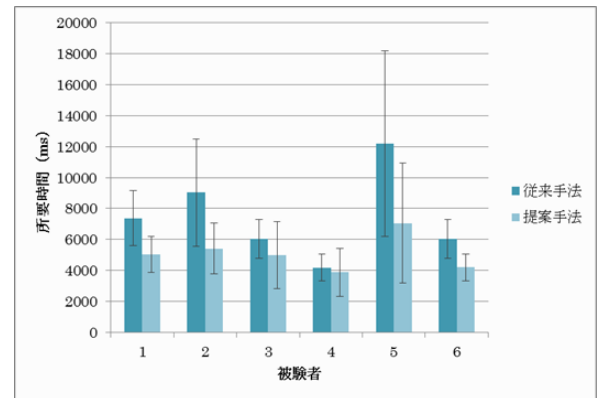


図 10 カーソル移動の被験者別平均所要時間

Fig.10 Average duration of cursor movement by participants

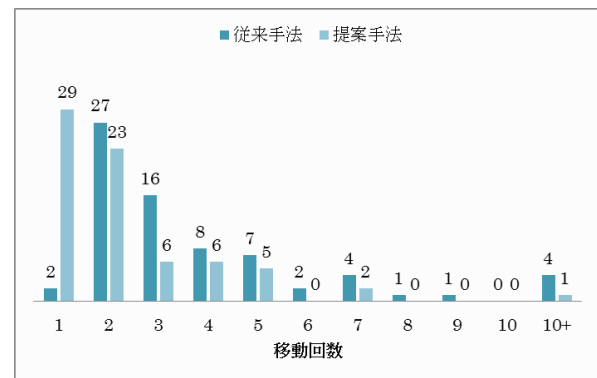


図 11 短距離移動回数の分布図

Fig.11 Histogram of numbers of relative movement

#### 4.3 考察

6 人の被験者における実験の結果を見ると、程度差こそあるが、被験者全員において、提案手法によるカーソル移動の所要時間が短縮されたことから、顔の向き方向を利用した本手法はポインティング操作の効率向上に有効であると推定できる。ただし、操作の平均所要時間は 4 秒以上と長くかかった。そこで、被験者の操作の様子を観察したところ、主に手によるカーソル移動に時間がかかっていると分かった。被験者の実験後のコメントによると、腕の伸び具合でカーソルの移動をコントロールするときに、カーソルの移動状態を示す有効なフィードバックがなく、被験者がカーソル移動の開始と停止のタイミングを把握できなかったことが一つの原因だと考えられる。カーソルの移動操作でのフィードバックを強化することで、操作の直感性が改善される事も考えられるため、今後の課題として実践すべきだと考えている。また、細かいカーソル移動を行うとき、ほとんどの被験者にはターゲットの所在位置に向いて手を伸ばすという動きが見られるが、この身体動作は疲労感を起こしやすいという問題があり、さらに移動操作にも不必要である。カーソル移動操作は掌を画面に向けて行われているため、手の向きがカーソル位置を決めているという被験者の誤

解が原因だと考えられる。ただし、練習とタスクの遂行を通して、この問題は軽減する傾向が見られた。

ターゲット選択毎の移動操作の回数についての実験結果を見ると、従来手法ではほとんどのタスクにおいて短距離の移動操作を2回以上行う必要があるが、提案手法では1回のみ移動操作を行うこと達成できるタスクが著しく多くなり、大体のターゲット選択が1〜2回の移動操作で達成できるようになった。短距離移動回数の減少は手による総移動距離の短縮を意味し、そして実験の過程からその要因が注目点移動による操作範囲の絞り込みだと見られる。この実験結果から見れば、顔の向き方向を用いたことがタスクの所要時間の短縮の重要な原因だと考えられる。また、手を使う操作の減少によりユーザの疲労感が軽減されることも被験者のコメントから分かった。これが操作効率の向上にも繋がると考えられる。

## 5. 関連研究

Kumar らによる EyePoint[4]では、キーボード操作により広範囲移動と局所の位置特定という二つの視線測定モードの切替えを行い、デスクトップにおける精密な視線操作を実現した。Blanch らが提案した Rake Cursor[7]では、Ninja Cursors[5]の複数カーソル操作手法を拡張し、注目点の位置を用いて複数カーソルから「アクティブカーソル」を選択することにより、マウスカーソルの移動の高速化を実現した。Porta ら[8]は視線測定を用いて直接カーソルを移動させる操作手法を考案した。これらの研究はいずれも視線測定と伝統的な操作方法を組み合わせることで操作の効率性と精密性を図っているが、デスクトップ環境での操作を対象としており、精度の高い視線測定を要求し、また、キーボードやマウス等のデスクトップ向けの操作機器を使用しているため、大画面操作に適切ではないと考えられる。

Nakanishi ら[2]は、顔認識による大画面操作手法を提案した。Vogel ら[3]は、大画面環境において、指差しによるカーソルの絶対位置移動と開けた状態の手の移動によるカーソルの相対位置移動を利用し、これらの手法によるポインティング手法を実装し、それぞれの手法とその組み合わせにおける性能について考察した。Schick らによる Extending Touch[6]では、大画面前方の空間の3Dモデルを構築することでユーザのジェスチャを認識し、指差し操作とタッチスクリーン操作を実現した。これらは本研究と同じ大画面環境における操作に関する研究であるが、本研究ではユーザの注目行為の利用に着目し、注目点の位置によって手の操作に存在する欠点を補い、マルチモーダルな入力により操作効率の向上を図っている。

## 6. まとめ

本研究ではユーザの顔の向き方向によってカーソル移

動の補助を行う大画面ポインティング手法を提案した。また、この提案手法の有効性を確認するためにプロトタイプシステムを実装した。このプロトタイプシステムでは擬似的な注目位置の認識と手の動きの認識により、速やか且つ正確な操作を可能にした。

プロトタイプシステムに対する評価実験を通して、提案手法の有効性が確認できた。一方で、手の動きによる操作に操作の直感性に関する問題点を発見した。今後の課題の一つとして、この問題点の細かい要因の分析と、手によるカーソル移動方法の改善を行うべきだと思われる。

多くの大画面は同時に複数のユーザに情報を提供するように設計されているため、画面がひとりのユーザにより独占されている場面が少ない。そのため、使用者が複数いる環境での提案手法の実現方法について検討する必要があると考えている。マルチユーザの認識やユーザ同士の操作の衝突などが解決すべき問題である。さらに、グループウェアとしてのインタラクションデザインも検討すべきである。

## 参考文献

- [1] Zhai, S.M., Morimoto, C., Ihde, S.: Manual and Gaze Input Cascaded (MAGIC) Pointing; In Proceedings of CHI '99, pp.246-253 (1999).
- [2] Nakanishi, Y., Fujii, T., Kitajima, K., Sato, Y., Koike, H.: Vision-Based Face Tracking System for Large Displays; In Proceedings of UbiComp 2002, pp.152-159 (2002).
- [3] Vogel, D., Balakrishnam, R.: Distant Freehand Pointing and Clicking on Very Large, High Resolution Displays; In Proceedings of UIST 2005, pp.33-42 (2005).
- [4] Kumar, M., Paepcke, A., Winograd, T.: EyePoint: Practical Pointing and Selection Using Gaze and Keyboard; In Proceedings of CHI 2007, pp.421-430 (2007).
- [5] Kobayashi, M., Igarashi, T.: Ninja Cursors: Using Multiple Cursors to Assist Target Acquisition on Large Screens; In Proceedings of CHI 2008, pp.949-958 (2008).
- [6] Schick, A., van de Camp, F., Ijsselmuiden, J., Stiefelwagen, R.: Extending Touch: Towards Interaction with Large-Scale Surfaces; In Proceedings of ITS '09, pp.117-124 (2009).
- [7] Blanch, R., Ortega, M.: Rake Cursor: Improving Pointing Performance with Concurrent Input Channels; In Proceedings of CHI 2009, pp.1415-1418 (2009).
- [8] Porta, M., Ravarelli, A., Spagnoli, G.: ceCursor, a Contextual Eye Cursor for General Pointing in Windows Environments; In Proceedings of ETRA 2010, pp.331-338 (2010).