

# ウィンドウインタフェースシステムにおける頭部位置・姿勢の有効な利用法

北島 光太郎<sup>†</sup>

佐藤 洋一<sup>‡</sup>

小池 英樹<sup>†</sup>

<sup>†</sup>電気通信大学 情報システム学研究科

<sup>‡</sup>東京大学 生産技術研究所

本稿では、画像処理を用いユーザの頭部位置・姿勢をリアルタイムに検出し、ヒューマンインタフェースとして有効に利用する手法を提案する。従来、視線をウィンドウインタフェースシステムへ応用した例では、視線の特徴からポインティングは高速に行う事が可能であるがコマンドの決定には不向きであった。本研究では頭部の位置・姿勢を有効に利用しウィンドウインタフェースシステムを操作する手法を提案する。それらの手法を評価するために X Window 上で顔の向きおよび位置を用いた操作を行うことができるシステムを試作し、基本的な操作に対する評価実験を行った。その結果から、頭部の位置・姿勢情報をインタフェースシステムにどのように利用することが有効であるかについて考察する。

## The Effective Use of Head Pose and Position for Window Interface System

Kotaro KITAJIMA<sup>†</sup>

Yoichi SATO<sup>‡</sup>

Hideki KOIKE<sup>†</sup>

<sup>†</sup> Graduate School of Information Systems, University of Electro-Communications

<sup>‡</sup> Institute of Industrial Science, The University of Tokyo

This paper discusses the use of head pose and position for human-computer interaction by using a real-time face tracking system. The previous studies indicate that the use of gaze information is not the best suited for executing commands while it can be used effectively for pointing tasks. In this work, we study how to use head pose and position more effectively for interactions in window-based interface systems. In particular, we propose the use of face directions in conjunction with other inputs, i.e., key inputs. We have conducted experimental studies by using our prototype system which can measure a user's face direction in real-time by using a vision-based face tracking without using any intrusive devices. Based on the experimental results, we demonstrate the effectiveness of the proposed approaches for using head pose and position for window-interface systems.

### 1. はじめに

現在、広く一般的に利用されているマウスやキーボードに代表されるインタフェースよりも快適に利用できるインタフェースとして、実世界指向、状況依存型インタフェース (PUI: Perceptual User Interface) に期待が寄せられている[1][2]。

PUI としては音声認識、ジェスチャ認識、視線認識などが研究されている。その中でも人間の目や顔の動きは人の注意や意図と深く関係していることや、高速に移動できる、疲れないといった特徴がある。そのため、ユーザの視線および顔の向きを検出することによってコンピュータによる状況認識及び、ユーザの意図理解への応用が期待できる。

これまでに視線情報をヒューマンインタフェースの分野に応用しようとする研究は数多く行われており、その多くがマウスやタッチパネルなどの入力デバイスを視線に置き換えたものであった。このようなシステムでは、コンピュータの操作すべてを視線で行おうとしているが、視線を向けたコマンドすべてが選択されてしまうという問題などにより、コンピュータインタフェースとして必ずしも効率的ではなかった[3]。また、視線計測時において頭部装着型のデバイスを利用した場合でも、測定誤差が生じてしまうことや人間の固視微動による誤差によって、ユーザが特定のドットを視線で指し続ける事は不可能に近い。そのため、上記のようなコマンド入力型のインタフェースシステムに視

線を利用する手法より、画面を見ている自然な注視点から情報を獲得する手法が望ましい[4]。

本稿では、画像処理によりリアルタイムに検出されるユーザの頭部位置および姿勢をウィンドウシステム操作に有効に利用する為の方法を検討する。具体的なシステムとして、X Window 上に組み込んだものを試作し、評価実験を行いその有効性を評価したのでここに報告する。

以下に本システムの特徴をあげる。

- ・ 画像処理を用い完全に非接触でリアルタイムにユーザの頭部位置・姿勢を検出することが可能。これによりユーザは特殊な装置を身につける事なく作業を行うことができる。
- ・ ユーザの顔の3次元姿勢をもとにX Windowのフォーカス制御、及びウィンドウの移動を行うことが可能。ユーザは作業を行いたいウィンドウに顔を向け、キーボードのキーを押すだけでウィンドウの選択、移動が可能となる。
- ・ 頭部の3次元位置・姿勢を用いてウィンドウ上でのアプリケーション操作が可能。
- ・ 提案システムはX Window上の環境への統合が容易であり、ユーザは簡単な操作方法を覚えるだけでより快適な操作環境が利用可能となる。

本稿ではまず第2章で実時間頭部位置・姿勢検出システムの構成を簡単に述べた後、第3章で今回提案する頭部の位置・姿勢によるインタフェースの枠組みを紹介し、提案手法を評価する為に用いたX Windowシステムの頭部位置・姿勢による制御手法について述べる。そして第4章で試作したシステムを用いた評価実験の結果を報告しその有効性について考察を行う。

## 2. 頭部位置・姿勢検出システム

従来、視線を検出するシステムとしては、頭部装着型のものと非装着型のものが開発されてきた。頭部装着型は、高速かつ安定に、視線を検出することが可能であるが、重量がありユーザが自然な動作を行うには適していない。一方、非装着型としては、カメラでユーザの顔領域を撮影し、画像処理を用いて視線を検出するシステムがある。これらのシステムでは、特徴領域を明確にするためにマーカーを顔につけるなどの制約や、システムのキャリブレーションに非常に時間がかかり使いにくいなどの問題点があった[5]。その為、日常のインタフェースシステムとして利用するには向いていないものが多かった。

インタフェースシステムとして快適に利用する為には、システムが完全に非接触、リアルタイム処理、受動的であることが必要である。本研究では松本らの手法を基本とし画像処理を用いた非装着型でカメラの前

に座るだけでリアルタイムに頭部の位置および姿勢の検出が可能なシステムを構築している[6]。

図1にシステム構成を示す。システムはユーザの顔領域を観察するカメラ2台と顔領域を照らす近赤外照明装置、画像処理を行うPC、処理結果を出力するモニタから構成される。

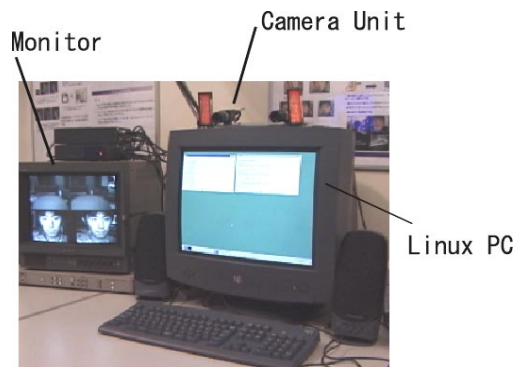


図1 プロトタイプシステム外観

本システムの基本的な処理の流れは、2台のカメラから複数の顔の特徴点をテンプレートマッチングにより検出し、頭部の3次元での位置・姿勢を決定するというものである。以下で処理の概略を説明する。

### (1) 初期化部

カメラユニットに取り付けられた近赤外照明の反射光をもとに顔領域を検出し、顔領域内で目と口の両端位置を検出する。検出した目及び口の両端をテンプレートとして登録する(図2(左))。この時、左右2台のカメラはあらかじめキャリブレーションしておく[7]。これにより、左右のカメラ画像中から得られた特徴点の対応を用いて、その特徴点の3次元位置が計算できる。このとき得られた特徴点の3次元位置を基準の剛体モデルとして登録しておく。

### (2) 主処理部

前フレームでの特徴点の3次元位置を基準に、フレーム間で特徴領域の移動量は微小であると仮定して2次元での特徴点の探索領域を決定する。このようにして得られた探索領域内においてテンプレートマッチングを行い、最も相関値の高い点を検出点とする(図2(右))。ただし、テンプレートマッチングで検出された点は誤差のため、間違った点が検出されてしまうことがある。そこでこれらの点は初期化部で登録された剛体モデルを元に剛体としての性質を保つように修正される。これにより、特徴点の一部が間違っても修正され、安定した検出が可能となる。最後に両目と口の特徴点の位置から定義される面を顔平面とし、その顔平面の法線方向から顔方向を推定する。ユーザが顔を向けている対象はこの法線ベクトルとディスプレイ平面の交点から求めることができる。

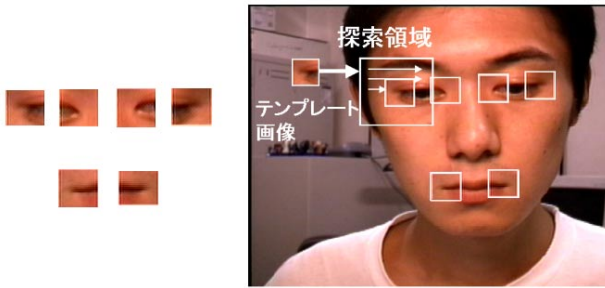


図2 テンプレート画像 (左) とマッチングの様子 (右)

上記の手法を用いて顔の特徴領域の追跡を行った結果を図3に示す。顔の移動に対しても、安定して目や口の両端の位置が検出できているのがわかる。また、テンプレートマッチングには正規化相関を用いている為、明るさの変化に対して比較的安定な追跡が実現されている。顔の姿勢に関しては、最大誤差約 $\pm 1$ 度、顔の位置に関して、最大誤差約 $\pm 1$ mmの精度での検出が可能である。また、処理速度に関してはPC (Pentium 550MHz, Linux OS) と HITACHI IP5005 画像処理ボードを使用し、毎秒 20 フレームでのリアルタイム処理が可能となっている。カメラは SONY XC-EI50 CCD-IR カメラモジュールを使用した。

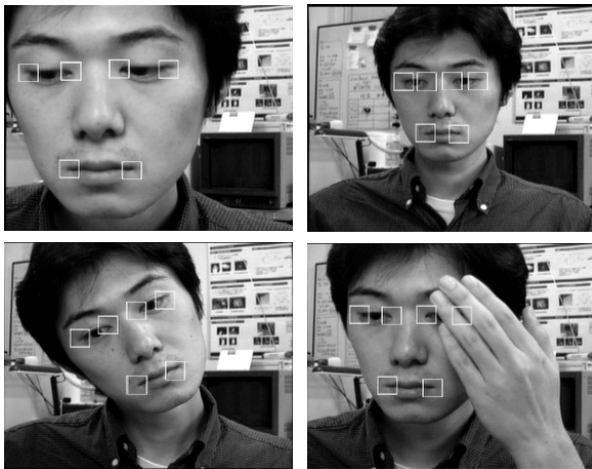


図3 トラッキングの様子

### 3. ウィンドウインタフェースにおける頭部の位置・姿勢の利用

我々が普段ウィンドウシステムを使用する際には、入力を行うウィンドウ (フォーカスウィンドウ) を切り替えながら作業を行っている。一般にフォーカスウィンドウを切り替えるときには、マウスポインタを目的のウィンドウの上に移動するか、またはキーボード操作によってフォーカスウィンドウを決定している。しかし、キーボードを主に利用している環境を考えた場合キーボードからマウスへの手の移動がユーザの負荷になってしまう。そこでユーザは作業を行うウィンドウを必ず見ている事から、視線検出システムを用い

ユーザが見ているウィンドウへ自動的にフォーカスが移動するシステムが提案されてきた。

これら従来のシステムでは、ウィンドウを見ている時間などを考慮してアクティブにするウィンドウを自動的に切り替えていた。例えば、大野のシステムでは、ユーザがウィンドウに注目している度合いを利用することで、アクティブにするウィンドウを決定している [8]。これにより、ウィンドウの境界付近を見ている時にたまたま見た他のウィンドウにフォーカスが移動してしまうという問題点を回避している。しかし、ユーザの自然な動作を見てみると、作業しているウィンドウを常に見ているわけではなく、手元にある本や隣のウィンドウを参照しながらキーボード入力を行うことが多い。また、ウィンドウ上のアプリケーションで、視線情報や頭部のジェスチャを利用する場合に、常に目的のウィンドウを見ながらジェスチャを行わなければならないという拘束があった。つまり、従来のシステムではユーザは入力を行う時に常に目的のウィンドウを見ながら作業することを強いられていた。

そこで本研究では、ウィンドウインタフェースにおける基本的な操作のうち、(1) フォーカスウィンドウの切り替え、(2) ウィンドウの移動、(3) ウィンドウの重なり制御、(4) ズームやスクロールなど 3 次元の操作の各々、を考えこれらの操作を頭部の位置・姿勢を用いてどのように効率よく行うことができるか検討する。また、これらの手法を評価するために、前章で述べた視線検出システムからの情報を利用して、X Window システムを制御するシステムを試作した。

これまでの研究で視線とマウスによる操作を組み合わせることで作業効率が上がることが報告されている [9][10]。これらは、視線の特徴である高速な移動とマウスによるユーザの明示的なコマンド決定を組み合わせたものであり、視線によるコマンド決定時のあいまいさをなくすことが可能となっている [3]。本手法では、顔の向きで位置の指定のみを行い、フォーカスの移動やウィンドウ移動のコマンド決定はユーザのキーボード入力にゆだねる。ここではユーザがキーボードを主に利用している環境を想定しているためキー入力に伴う負荷は少ないと思われる。またキーが押された時の頭部姿勢のみを利用する為、頭部姿勢検出システムの誤差による影響の軽減が期待できる。視線ではなく顔の向きを用いた理由は、画像解像度や視線の特徴の問題から視線方向を検出すると顔の向き以上に誤差が大きくなりディスプレイ上の点を高精度で指定することが難しくなる為である。顔の向きを用いた場合、高速な眼球の動きを利用できなくなるため視線を用いた時のような高速性は期待できないが、キーボードからマウスに手を移動するよりは作業効率が上が

と思われる。また、頭部の3次元位置をズームやスクロールなどの3次元操作に利用することにより、マウスなど2次元のポインティングデバイスには不向きな操作をより直感的に行うことが可能になると期待される。以下で各操作方法について述べる。

### 操作1：フォーカスウィンドウの切り替え

システムによりユーザの頭部姿勢はリアルタイムに検出される。ユーザによってキーボードのフォーカス制御キーが押されると、ユーザの顔が向いているウィンドウにフォーカスが移動する。ウィンドウが複数枚重なっている時には一番上のウィンドウにフォーカスが移動する。この時、一度フォーカスが移動するとフォーカスは固定されるので、各ウィンドウアプリケーション上で独立して頭部による操作が可能となる。

### 操作2：ウィンドウの移動

ウィンドウが複数枚重なっている時など、ユーザは頻繁にウィンドウを移動する。マウスなどを使用した場合、ウィンドウの移動はウィンドウのタイトルバーをドラッグすることで可能である。しかし、その場合キーボードからマウスへ手を移動することがユーザにとって負担となっていた。また、従来の視線を使用したウィンドウ制御システムでは、コマンドを選択するなどしてウィンドウの移動を行っていたが、コマンド選択にかかる時間や疲労、移動場所の決定時などに問題があり、必ずしも快適な操作環境ではなかった。

これに対し本手法では、ウィンドウ移動キーを押すことで、現在フォーカスのあるウィンドウを顔が向いている位置に移動する。これは顔を他の方向に動かしてもフォーカスウィンドウが自動的に変更されない為可能となっている。

### 操作3：ウィンドウの重なり制御

大きなウィンドウがディスプレイ最前部にある為に、作業を行いたいウィンドウが完全に隠され、見えない場合がある。そのような時には、最前部のウィンドウを他の位置に移動するかアイコン化するなどして、目的のウィンドウを最前部に移動する。本手法では重なり制御キーを押すことで、現在のフォーカスウィンドウを最後部に移動する。この時、後部のウィンドウが自動的に最前部に移動しフォーカスも後部のウィンドウに移動する。

### 操作4：ズームやスクロールなどの3次元操作

ウィンドウアプリケーションにおいて、3次元の操作が必要なものは多く存在する。例えば、地図アプリケーションやドローツールでは頻繁に拡大率を変更しながら作業している。また、3次元空間におけるウォークスルーでは必然的に3次元の操作が必要になってくる。これらの操作をマウスなど2次元の入力デバイスで行うにはモードの変更やキー入力の組み合わせを必要とする為、直感的でなく操作が複雑になるなどの

問題点があった。そこで頭部の3次元位置・姿勢を利用することで3次元操作をより直感的にかつ高速に行う事ができる手法について地図アプリケーションを例に検討する。地図のアプリケーションを利用する場合には、縮尺率を変更せずに目的の場所を探すよりも、縮尺率を変更しながら探す方が効率が良いことが知られている[11]。しかし、縮尺率を変更するためにはモードの変更や画面を切り替える必要があり、目的の縮尺率にすぐに調節することは困難であった。そこで、頭部の位置を利用して地図の拡大縮小を行い、地図の縮尺率を瞬時に変更できるシステムを試作した。操作はウィンドウ上に地図が表示されており、基本的には図5に示すようにユーザがディスプレイに顔を近づけるとズームイン、遠ざけるとズームアウトする。また顔の向きによって画面のスクロールを行うことができる。今回は、頭部の位置を用いズームレベルを制御する3種類の手法を評価した。

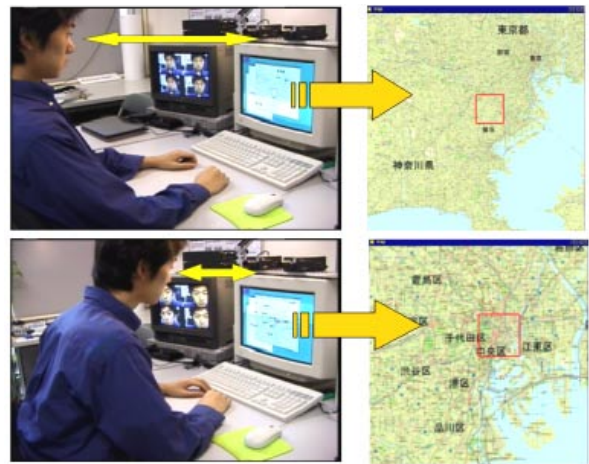


図4 地図アプリケーションの操作

#### ・ 手法1

頭部の位置が直接ズームの割合になっている。つまり、ディスプレイに顔を近づけると地図が拡大して表示され、顔を遠ざけると縮小して表示される。(ダイレクト型)(図5)。



図5 手法1(ダイレクト型)

#### ・ 手法2

ジョイスティックのアナロジーに基づき頭部の位置を使用しズームレベルを操作する。自然な姿勢で頭部がある位置をニュートラル位置とする。ある閾値以上ディスプレイに近づくと on となり、一定の速度でズー

ズームし続ける．再びニュートラル位置に頭部を戻すとズームは止まる（デジタル型）（図6）．

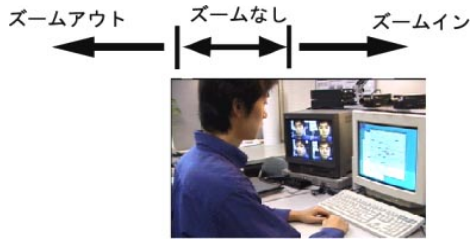


図6 手法2（デジタル型）

・ 手法3

マウスボタンを押している間のみダイレクト型のように操作することが可能である．このためボタンを押している時以外は頭部を自由に動かすことができる．また，ユーザがマウスのボタンを押した時の頭部の位置を基準にズームの原点がシフトする為，マウスボタンを押した時に突然ズームレベルが変化することがない．（ダイレクト+原点シフト型）（図7）．

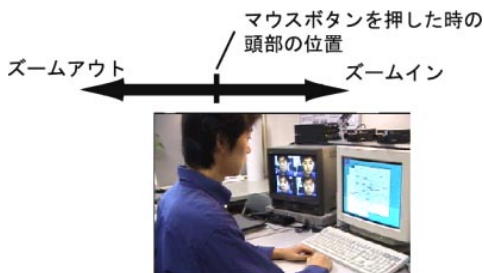


図7 手法3（ダイレクト+原点シフト型）

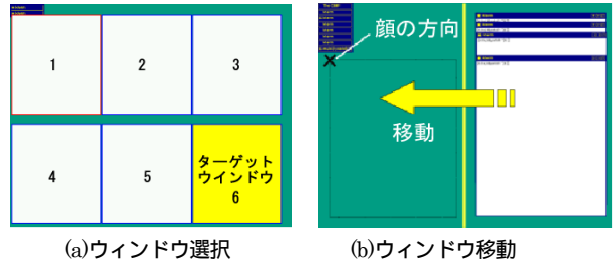
4. 評価実験

以上で述べた手法の有効性を評価するために，ウィンドウ操作及び，アプリケーション操作に関する実験を行った．それぞれの実験において被験者は5人である．また，被験者は日頃 Window システムを利用しており，マウスやキーボードの操作に慣れている．ディスプレイは19インチのものを使用した．

4.1 ウィンドウ選択に関する実験

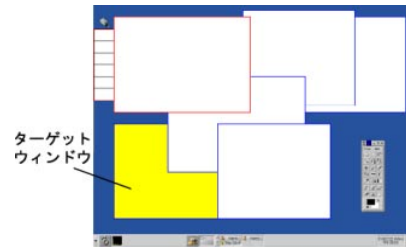
まず，ウィンドウ選択にかかる時間を顔の向き，マウス，キーボードの各手段を用いて測定した．ここではユーザがプログラム開発など基本的にキーボードを利用して作業を行っている状況を想定した上で入力手段とウィンドウの枚数の違いによりウィンドウ選択時間にどのような差が生じるのか比較を行う．ディスプレイ上にウィンドウが重ならない状態で2枚，4枚，または6枚現れる．この時ウィンドウが重ならないようにするため，枚数が増えるとウィンドウのサイズが小さくなる．表示されたウィンドウ中の1枚が緑色に変化する（ターゲットウィンドウ）．タスクとして，被験者は顔の向き，マウス，キーボードのそれぞれの入力手段によりターゲットウィンドウにフォーカスを合

わせた後スペースキーを押し選択する．正しいウィンドウを選択した場合，ターゲットウィンドウは他のウィンドウにランダムに移動する．被験者には10回連続でターゲットウィンドウを選択してもらいその時間を測定した．実験は各入力手段に関して5セット行いその平均を用いている．図8(a)にウィンドウが6枚の時の様子を示す．また，各入力手段のウィンドウ制御は以下に示す通りである．



(a)ウィンドウ選択

(b)ウィンドウ移動



(c)ウィンドウ重なり制御

図8 ウィンドウ選択実験環境

・ 顔の向き

被験者は目的のウィンドウの方向を向き，フォーカス移動キーを押すことで，目的のウィンドウにフォーカスを移動することができる．

・ マウス

マウスカーソルを目的のウィンドウ上に移動することでフォーカスを移動することができる．ただし，本実験では，被験者が普段キーボードによる操作を行っており，フォーカスの移動時のみマウスを利用する場合を仮定しているため，片手のみで操作を行ってもらった．また，手の移動にかかる時間も一定にするために，マウスを置く位置はキーボードの右側に固定した．

・ キーボード

Ctrl キーを押すことでフォーカスウィンドウを巡回させることができる．

図9に各入力手段についてウィンドウを10回選択するのに要した時間を示す．ここで，グラフの縦線は標準偏差を示している．顔の向きはウィンドウの枚数に関わらず，安定して高速なウィンドウ選択を行うことができた．ウィンドウが2枚の場合にはキーボードを使用した時が最も速いが，これはフォーカスを移動する対象のウィンドウが1枚しかないため，単純にキーを一回押すだけでターゲットウィンドウにフォーカスを移動できるためである．反対にウィンドウの枚数

が増加するにつれてキーボードによる操作は時間が増加している。マウスはキーボードからマウス位置への手の移動時間が長いため時間がかかっていると考えられる。しかし、ウィンドウの枚数が増えても、あまり影響を受けず安定した操作が可能であることを示している。

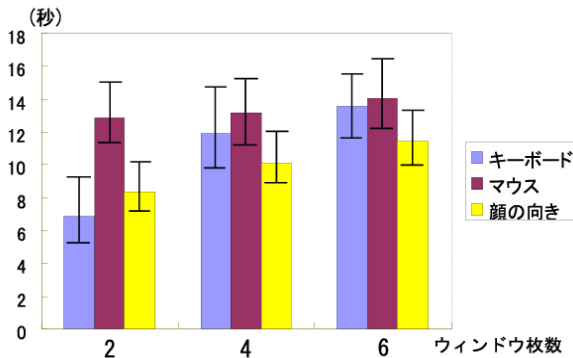


図9 ウィンドウ選択時間結果 (10回分合計)

次に、ウィンドウが重なっていない場合において、ユーザが実際にキー操作を行う回数について考察する。フォーカスをターゲットウィンドウに移動するためには、顔の向きを利用した場合、1回のキー入力が必要になる。キーボードを利用した場合には、ウィンドウが2枚の時には1回の入力済むが、ウィンドウの枚数が増加するにしたがって入力回数は増えていく。ウィンドウが6枚の時には最大5回のキー入力が必要になってくる。設定によって逆向きの巡回を行うこともできるがその場合でも、最大3回のキー入力が必要になる。つまり、ウィンドウの数が4枚以上の場合には顔の向きを用いた方が少ない操作で作業を行えるということが言える。マウスはキー操作を伴わずにウィンドウ選択が可能であるため、単純に操作回数での比較はできないが、キーボードからマウスへ手を移動する際の時間や疲労を考えると、ウィンドウの選択において、顔の向きはマウスよりも効率的であると言える。一方でアイコンの選択など高精度でポインティングを行わなければならないようなタスクでは、顔の向きによるポインティング精度の低さによりマウスの方が適していると思われる。また、今回はウィンドウの選択時において基本的にキーボードをメインで使用している場面を想定しているが、ウェブのブラウジングのようにマウスをメインで使用するような場面では、視線とマウスとの組み合わせによる操作も有効であると思われる[9]。

#### 4.2 ウィンドウの移動に関する実験

次に、ウィンドウの移動にかかる時間をマウスと顔の向きで比較した。図8(b)に示すように、画面の片側にウィンドウが10枚あり、そのウィンドウをすべて画

面の反対側に移動する時間を計測する。各手法でのウィンドウ移動方法について以下で述べる。

##### ・顔の向き

移動したいウィンドウにフォーカスを合わせた後、移動したい場所を向きウィンドウ移動キーを押す。

##### ・マウス

ウィンドウのタイトルバーをドラッグして移動する。

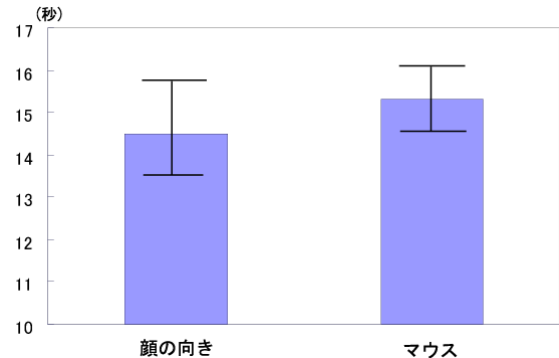


図10 ウィンドウ移動時間結果 (10回分合計)

各被験者に対して5セットの計測を行った結果が図11である。時間に関しては、若干視線の方が早かったが、大きな差は見られなかった。前節の実験からも分かるように、マウスを使用する場合、キーボードからマウスへの手の移動時間が大きな問題となる。しかし、今回の実験ではマウスから手を離さず作業を行うことができたため、差が生じなかったものと思われる。実際にキーボードを使用する作業においては、キーボードから手を離さずウィンドウの移動を行える顔の向きは有効であると思われる。

#### 4.3 ウィンドウの重なり制御に関する実験

図8に示すような、ウィンドウが複数枚重なっている状態でターゲットウィンドウを選択するのに要する時間を測定した。ターゲットウィンドウが最上位のウィンドウに隠されている場合、顔の向きを利用した場合は重なり制御キーを押すことで現在フォーカスが存在するウィンドウを最後部に移動することができる。一方、キーボードを利用する場合には、ウィンドウ選択の実験と同様にCtrlキーを押すことですべてのウィンドウを順に巡回する。マウスを使用する場合、最上位のウィンドウを他の場所へ移動する、アイコン化する、またはコマンド選択によって最下位に移動する、などさまざまな方法で制御行っており単純に比較できない。そのため、今回は視線とキーボードの比較のみを行った。

実験は基本的に4.1節のウィンドウ選択に関する実験と同様に行い、ウィンドウが重なった状況でターゲットウィンドウを10回選択するのに要する時間を測定するという作業を5セット行った。

ウィンドウの枚数を4枚、6枚、または8枚と変え

た時の結果を図 1 1 に示す．ウィンドウの枚数に関わらず，顔の向きを利用した方が短い時間でウィンドウを選択できるという結果になった．ただし，ウィンドウの総数が少ない場合には，4.1 節のウィンドウ選択の実験から分かるようにキーボードによってフォーカスを巡回する方法が効率的であると思われる．しかし，ウィンドウの枚数が多くなると，キーボードによる選択は効率が悪くなる．本手法では，ウィンドウの重なり制御キーを用いることで顔が向いている場所の付近のみでウィンドウのフォーカスを巡回することができるため，すべてのウィンドウに対してフォーカスが巡回するキーボードと比較した場合も効率的であると思われる．

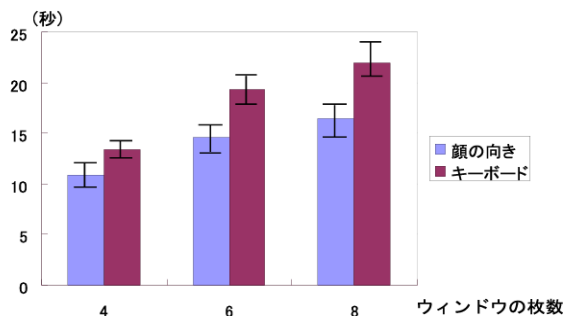


図 1 1 ウィンドウ重なり実験結果 ( 1 0 回分合計 )

#### 4 . 3 頭部の位置によるアプリケーション操作

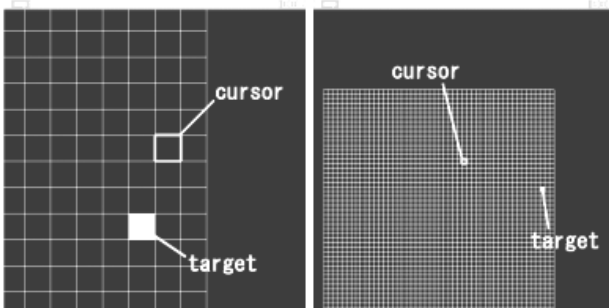


図 1 2 地図アプリケーション操作実験

次に，頭部の位置によるズーム操作の有効性を検証するための実験を行った．実験では図 1 2 に示すように画面に 5 0 × 5 0 のセルが表示される．このうち 1 つのセルが黄色に変化する (ターゲット)．被験者は，ズーム機能を利用しターゲットを発見し，カーソルをマウスで操作しターゲットに重ね，マウスのボタンを押すことで選択することができる (このときターゲットに対して十分ズームインしていないと選択できない)．ターゲットは選択されると現在の場所から一定の距離離れた場所にランダムに移動する．被験者にはターゲットを 1 0 回連続して選択してもらいその時間を測定した．測定は各手法について 5 回である．ズーム操作には頭部の位置を利用した手法 1 (ダイレクト型)，

2 (デジタル型)，3 (ダイレクト+原点シフト型) とキーボードを用いたタスク終了までの時間を測定した．キーボードはズームインとズームアウトを行うそれぞれのキーを用意した．また，カーソルの移動にはすべてマウスを使用している．結果を図 1 3 に示す．

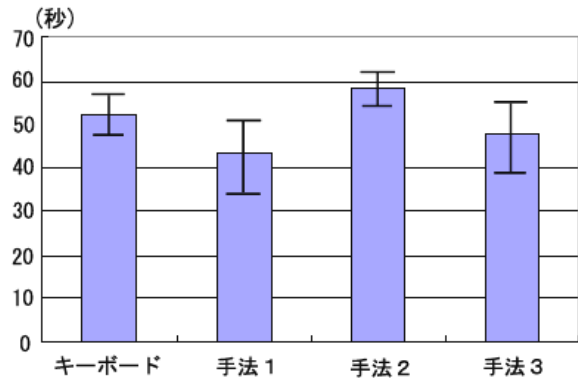


図 1 3 頭部の位置による操作実験結果 ( 1 0 回分合計 )

手法 1 (ダイレクト型) が最も高速であるという結果になった．これは，頭部の位置が直接ズームの割合に対応している為，ズームにかかる時間を省略することができる為だと思われる．また，地図全体の確認 (ズームアウト) と，詳細の表示 (ズームイン) を最も直感的に行う事ができた．しかし，その縮尺率のまま操作するには頭部の位置を固定したまま作業しなければならない．つまり，極端にディスプレイに近づいた状態や遠い状態で作業しなければならない場面が生じるため不便であった．そこで，本手法は空間上の情報を常に移動しながら眺めるような作業に向いていると思われる．例えば，画像ブラウザなどにおいてさまざまな画像を眺めながら好きな画像を探し出すような作業は本手法に向いていると思われる．

手法 2 (デジタル型) は，最も時間がかかったが，ズームの速度を変更することで，時間を短縮することが可能であると思われる．しかし，ズームの速度を速くしすぎると，微調整ができず，ユーザが希望する縮尺率にすることが難しくなってしまう．ニュートラル領域からの移動量に応じてズーム速度を変更することで微調整は可能になるが，結局，作業を行う時には頭部をニュートラル位置に固定することが必要となりそれがユーザの負担になってしまうという結果になった．

手法 3 (ダイレクト+原点シフト型) は比較的高速に操作が可能であり，ズームの微調整も可能であった．また，被験者全員が手法 3 を最も使い易いと答えた．これは，ボタンを押しているとき以外は頭部の位置が自由であり，また直感的でもあることが理由として考えられる．他の手法では顔の位置がある程度拘束される為，疲れるという意見も多かった．この手法は普段は画面を固定している状況で作業し，場合によってズ

ームレベルを変更するようなタスク，例えばドロースケールなどに対して有効であると思われる。

#### 4.5 被験者の主観的評価

実験後にアンケート記入形式での質問を被験者に対して行い，頭部の位置と顔の向きによる操作の主観的な使いやすさについて調べた。アンケートの項目は各操作に関して使いやすさ，精度，疲れ，慣れ，有効性，好み，などを含んでいる。

まず，顔の向きによるウィンドウの各操作については，その有効性や好みに関して良いという意見が多かった。逆に，疲れの項目はマウスやキーボードよりもやや疲れるという意見が多かった。これは顔トラッキングシステムの精度の問題から被験者がある程度顔の姿勢を保ったまま実験タスクを行ったことが原因と思われる。これは今後の顔トラッキングシステムの精度向上によって解決するものと思われる。また，視線でなく顔の向きを利用することによるギャップに関しては，ディスプレイのように狭い対象範囲では顔をやや意識して動かす必要があり若干の違和感があるという意見があった。つまり顔の向きを利用する場合には大型のスクリーンのように視線に対応して自然と顔が向く大きさの対象の方が自然であると思われる。

次に，頭部の位置によるズーム操作についてはその有効性や好みにおいて高い評価を得ることができた。特に他のデバイスと比較して直感的であるという意見が多くみられた。疲れに関しても他のデバイスと同程度であり，頭部を動かす負担をあまり感じなかったという意見が多くみられるなど，頭部の位置による操作は顔の向きによるインタラクションよりも全体的に高い評価を得ることができた。

#### 5.まとめと今後の課題

ウィンドウインタフェースシステムにおいて頭部の位置・姿勢の情報を有効に利用するために，ウィンドウシステムの各基本操作について頭部の姿勢および位置の有効な利用法について検討を行った。具体的には，ウィンドウシステムの4つの操作について頭部の位置・姿勢を利用した手法を提案した。ここで共通したアプローチとして，顔の向きをポインティングのみに利用し，コマンドの決定部分はユーザに委ねることにした。また，3次元の操作を行うタスクには頭部の位置情報を用いた。これらの手法を評価するために，画像処理による頭部の位置及び姿勢が検出可能なシステムを利用しX Windowの制御及びアプリケーション操作を行うシステムを試作し，その使いやすさについての評価を行った。

ポインティング作業は，ユーザが必要な時のみ行う事ができるため，従来のシステムにあったストレスを

感じる事がなく，また，他のデバイスよりも高速に作業を行うことができた。また，頭部の位置による操作は，3次元の操作を必要とするタスクに対して，マウスなど2次元のポインティングデバイスを利用するよりも直感的で効率が良いと言える。さらに，頭部位置・姿勢検出には頭部装着型のデバイスを用いていないため，ユーザはカメラの前に座るだけでシステムの利用が可能となった。

今後の課題としては，今回提案した手法のより詳しい評価，視線情報をさらに有効に利用するためのインタフェースデザインの検討，また，視線検出部における課題としては，視線検出の精度の向上，顔の姿勢と瞳の位置からより細かな視線方向を測定すること，顔トラッキングシステムの安定性の更なる向上などがあげられる。

#### 参考文献

- [1] M. Turk, "Moving from GUIs to PUIs," *Proc. Fourth Symposium on Intelligent Information Media*, Tokyo, Japan, December 1998.
- [2] 暦本純一，長尾確，"ポストGUI: 今後の展望"，Bit別冊ビジュアルインタフェース-ポストGUIを目指して-，共立出版 pp.177-198, 1996.
- [3] R. J. K. Jacob, "What you look at is what you get: Eye movement-based interaction techniques," *Proc. CHI'90*, pp.11-18, 1990.
- [4] R. J. K. Jacob, J. J. Leggett, B. A. Myer, R. Paucsh, "Interaction styles and input/output devices," *Behavior and Information Technology*, Vol.12, No.2, pp. 69-79, 1993.
- [5] 伴野明，岸野文郎，小林幸雄，"動向の抽出処理と頭部の動きを許容する視線検出装置の試作"，電気情報通信学会論文誌 D- Vol.J76-D- No.3, pp. 636-646, 1993.
- [6] Y. Matsumoto, A. Zelinsky, "Real-time Face Tracking System for Human-Robot Interaction," *Proc. 1999 IEEE International Conference on Systems, Man, and Cybernetics (SMC '99)*, pp. 2-830-2-835, 1999.
- [7] R. Y. Tsai, "A Versatile Camera Calibration Technique for High Accuracy Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses," *IEEE Journal of Robotics and Automation*, Vol. 3, No. 4, pp. 323-344, August 1987.
- [8] 大野健彦，"視線を利用したウィンドウ操作環境"，信学技報 HIP99-29, pp. 17-24, 1999.
- [9] S. Zhai, C. Morimoto, and S. Ihde, "Manual and Gaze Input Cascaded (MAGIC) Pointing," *Proc. CHI '99*, pp. 246-253, 1999.
- [10] 大和，神代，門田，松本，井上，"視線によるマウスカーソルの自動移動"，情報処理学会研究報告 99-HI-84, 1999.
- [11] G. W. Furnas and B. B. Bederson, "Space-Scale Diagrams: Understanding Multiscale Interface," *Proc. CHI '95*, pp. 234-241, 1995.