視線・マウス併用型インタフェースのドラッグ&ドロップ操作への適用

大和正武[†,☆] 神代知範[†,☆☆]
門田暁人[†] 松本健一[†]

本稿では，一般的なGUI上でのドラッグ&ドロップ操作の操作効率の向上を目的として，ユーザの視線の動きとマウスを併用した操作方式を提案する．提案方式ではドラッグ&ドロップに含まれる操作をドラッグアイコンとドロップアイコンの2つのアイコンを連続して選択する操作であると見なし，それぞれのアイコンの選択について，(1) 選択対象であるアイコン付近への大まかなカーソルの移動，(2) アイコンへのカーソル位置の微調整，(3) アイコン選択の確定操作，の3つの基本操作に分け，(1) には視線を，(2)，(3) にはマウスを用いる．微調整にはSemiAuto方式[15]を用いる．SemiAuto方式を用いることにより視線追跡装置の計測誤差や固視微動があっても，一般的なGUI上で目的のアイコンを指し示すことができる．提案方式を一般的なGUIに似せた環境で評価した結果，提案方式による操作は従来のマウスのみの操作に比べて，操作誤りを大幅に増やすことなく，1回の操作について平均で約17%（0.4秒）短く選択できた．また初心者を想定して（利き腕でない）左手を用いた実験では，操作誤りも減り，平均で約27%（0.8秒）短く選択できた．

Applying Gaze-added Interface to Drag-and-Drop Opeation

Masatake Yamato,[†,☆] Tomonori Kumashiro,[†,☆☆]
Akito Monden[†] and Ken-ichi Matsumoto[†]

The purpose of this paper is to increase the efficiency of drag and drop operation under the general GUI environment. For this purpose, we propose an operation method employs both eye gazing and a mouse together. In the method we regard drag and drop operation as a sequence of two icon selections; drag-icon and drop-icon. We split the each icon selection into 3 sub operations: (1) rough cursor movement toward the icon, (2) cursor position adjustment onto the icon, and (3) pressing/releasing the icon. We use eye gazing for (1), and, a mouse for (2) and (3). As a method for the cursor position adjustment, we use Semi Automatic method [15]. With using Semi Automatic method, the user can select an icon even if jittery motions of user's eye and the measurement error of an eye-tracking device occurred. We conducted an experiment to evaluate the proposed method on a GUI modeled on a general GUI. The result showed the proposed method is about 17% (0.4 sec.) faster than the mouse only operation per one drag and drop operation in average; and the error rate of the proposed method is almost the same as the mouse only operation. Furthermore, with using non-preferred hand, which is known to behave like a beginner's mouse operation, the error rate of the proposed method is smaller than a mouse only operation; and the proposed method is about 27% (0.8 sec.) faster than the mouse only operation in average.

1. はじめに

近年，人間の視線を計算機への入力に利用した「視線インタフェース」の研究がさかんに行われている．その主な研究目的の1つは，文字入力およびボタンやアイコンなどのターゲット選択を視線によって行う手段を提供することで，肢体不自由者のコミュニケーションを支援することである[6),9)]．このような目的を持つ視線インタフェースは特にGaze-Centered Interfaceと呼ばれる[10)]．マウスやキーボードなどを用いる従来のインタフェースと比べると必ずしも効率は良くないが，手が使えない状況では大変有用である．

視線インタフェースのもう1つの研究目的は，マウスやキーボードなどの従来の入力デバイスと視線を併

† 奈良先端科学技術大学院大学情報科学研究科
Graduate School of Information Science, Nara Institute of Science and Technology
☆ 現在，レッドハット株式会社
Presently with Red Hat Japan, Inc.
☆☆ 現在，TIS株式会社
Presently with TIS Inc.

用することで，より効率の良いインタフェースを実現することである．このような併用型のインタフェースはGaze-Added Interfaceと呼ばれる[10]．視線の移動速度はきわめて速く，眼球から50 cm前方に位置する21インチディスプレイの対角上を端から端まで移動する場合でも150 msec程度の時間しか要しない[7]．したがって，マウスカーソル（カーソル）をGUI部品へ移動させる操作（ポインティング操作）のために視線を入力として追加することは，インタフェースをより効率の良いものに発展させるうえで有望である[14),15)]．

本稿では，MS-WindowsやMacOSなどの一般的なGUI上で日常的に行われる操作のうち，ドラッグ&ドロップ操作を対象としたGaze-Added Interfaceの実現を目的とする．従来は，視線を用いるために特殊化されたGUI上でのGaze-Added Interfaceが主に研究されてきたが[3),10)]，現在普及している一般的なGUI上でのGaze-Added Interfaceが実現できれば，既存のアプリケーションソフトウェアのGUIを新規に作り直す必要なく操作を効率化できるため，多くの計算機ユーザにとってより有用であると考えられる．一般的なGUI上の操作には，アイコンやメニューなどのターゲット選択，ドラッグ&ドロップ，ウィンドウのスクロール，ウィンドウの選択など多数あるが，従来研究では，主にターゲット選択操作を対象としたGaze-Added Interfaceが提案されてきた[10),17)]．一般的なGUI上の操作をさらに効率化するためには，ターゲット選択操作以外のGUI操作に対しても視線の適用範囲を広げていくことが重要である．ドラッグ&ドロップ操作は多くの計算機ユーザが日常的に行うGUI操作の1つであり，かつ，従来研究されてきたターゲット選択操作との類似点が多いことから，本稿での研究対象に選んだ．

しかし，視線用に特殊化されたGUIと比較すると，一般的なGUIでのGaze-Added Interfaceの実現は容易でない．視線用に特殊化されたGUIでは大きなGUI部品（アイコン，メニューアイテムなど）が広い間隔で設置されているのに対し，一般的なGUIでは，小さなGUI部品が狭い間隔で配置されているため，次に示す視線追跡装置の計測誤差の問題，および，固視微動の問題を克服することが困難となる．

視線追跡装置の計測誤差 ユーザが見ている画面上の場所（注視点）を正確に特定することは，現在の視線追跡装置では困難である．視線追跡装置の測定値とユーザが実際に見ている画面上の場所との間には0.9 cm〜1.7 cm程度のずれがある（眼球から50 cm前方に計算機ディスプレイが位置する場合）[7),12)]．

固視微動 眼はつねに微動している．注視点は0.4 cm四方の範囲で細かく振動している．視線で1点を指示し続けることはできない（眼球から50 cm前方に計算機ディスプレイが位置する場合）[2)]．

本稿では，計測誤差と固視微動の問題を克服するために，文献15)で提案されているターゲット選択方式（SemiAuto方式）をドラッグ&ドロップ操作に応用する．SemiAuto方式は，狭い間隔で配置された小さなターゲットを正確にかつ効率良くポインティングできるように，ターゲットの近傍でのカーソル位置を自動的にもしくは手動で微調整できる．本稿で提案する方式は，ドラッグ&ドロップ操作を，ドラッグの対象となるアイコン（ドラッグアイコン）とドロップ先のアイコン（ドロップアイコン）の選択という連続する2回のターゲット選択操作であると見なし，それぞれの選択操作に対してSemiAuto方式を適用する．これにより，各選択操作を正確にかつ効率良く行うことが可能となる．

本稿では，提案する方式を一般的なGUI環境に似せた環境で行った評価実験の結果についても述べる．詳しくは4章で述べるが，従来提案されているGaze-Added Interfaceは，GUI部品の配置間隔を広くとるなどした特殊化したインタフェース上でしか評価されておらず，一般的なGUIでの有効性が明らかにされていない．本稿では評価実験により，一般的なGUIでの提案方式の効率と正確性を調べる．

以降，2章で方式を提案する．3章で提案方式の評価実験について述べる．4章で関連する研究について述べ，5章でまとめる．

2. 提案方式

提案方式について述べるにあたり，まず2.1節において，従来のマウスのみによるドラッグ&ドロップ操作について整理する．2.2節では，ドラッグ&ドロップ操作に視線を割り当てた単純な方式（基本方式）について述べる．この基本方式は提案方式の基本となる考え方を示すためのものであり，視線追跡装置と固視微動の問題に対して特別な工夫を行っていない．次に2.3節において，基本方式をベースにしてSemiAuto方式[15)]を応用した方式（拡張方式）について述べる．この拡張方式が本稿における提案方式である．

2.1 マウスのみを用いたドラッグ&ドロップ操作

ドラッグ&ドロップ操作は次に示す4つのより細かい操作の系列である[5)]．

(1) ポインティング操作1：ドラッグアイコンへカー

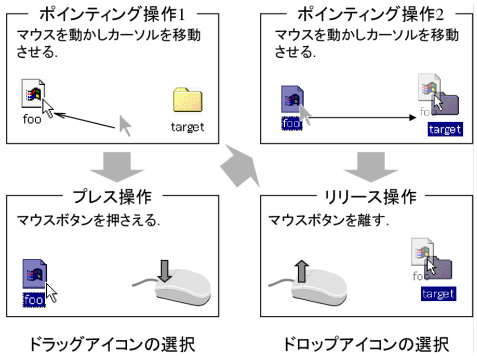


図 1 マウスのみを用いたドラッグ&ドロップ操作
Fig. 1 Drag and drop operation with mouse only.

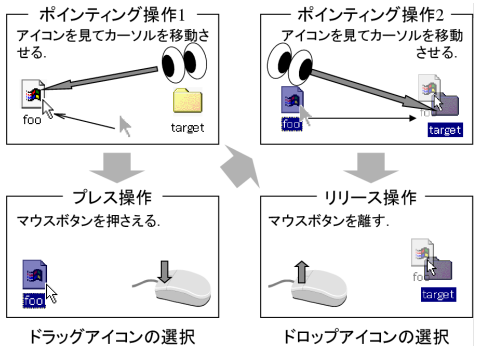


図 2 基本方式
Fig. 2 Basic method.

ソルを移動する操作（**図 1** 左上参照）

(2) プレス操作：マウスボタンを押してドラッグアイコンの選択を確定する操作（図 1 左下参照）

(3) ポインティング操作 2：ドロップアイコンへカーソルを移動する操作（図 1 右上参照）

(4) リリース操作：マウスボタンを離してドロップアイコンの選択を確定する操作（図 1 右下参照）

以降では，プレス操作とリリース操作をまとめて確定操作と呼ぶこととする．

2.2 基本方式

基本方式では，ドラッグ&ドロップ操作を構成する基本操作のうち，ポインティング操作を視線で行う．カーソルはユーザの画面上の注目箇所（注視点）に追従し移動させる．確定操作は従来どおりマウスボタンで行う（**図 2** 参照）．したがって，ユーザは次に示す細かい操作の連続によってドラッグ&ドロップ操作を行うことができる．

(1) ポインティング操作 1：選択したいドラッグアイコンを見る（図 2 左上参照）．

(2) プレス操作：マウスボタンを押してドラッグアイコンの選択を確定する（図 2 左下参照）．

(3) ポインティング操作 2：選択したいドロップアイコンを見る（図 2 右上参照）．

(4) リリース操作：マウスボタンを離してドロップアイコンの選択を確定する（図 2 右下参照）．

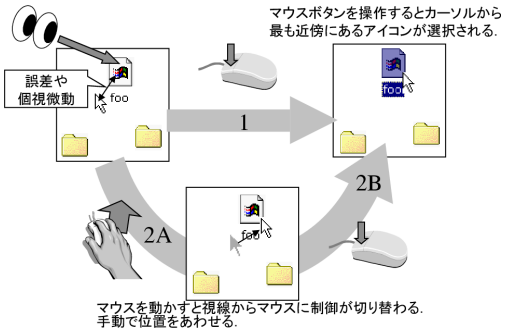


図 3 SemiAuto 方式（ドラッグアイコンの選択の場合）
Fig. 3 SemiAuto method (drag-icon selection).

2.3 拡張方式

拡張方式では，視線追跡装置の計測誤差や固視微動があってもカーソルを正確にアイコンへ移動させることができるように，インタフェースが自動であるいはユーザが手動でカーソル位置の微調整を行う．おおまかなポインティング操作および確定操作については，基本方式と同様にそれぞれ視線とマウスボタンを用いる．

微調整は 1 回のドラッグ&ドロップ操作において 2 回行う．ドラッグアイコン選択操作時（ポインティング操作 1 とプレス操作）とドロップアイコン選択操作時（ポインティング操作 2 とリリース操作）の 2 回である．微調整には文献 15) においてターゲット選択方式として提案されている SemiAuto 方式を用いる．ドラッグアイコンの選択とドロップアイコンの選択を 2 回のターゲット選択操作であると見なし，各選択において SemiAuto 方式によりカーソル位置の微調整を行う．

SemiAuto 方式による微調整を**図 3** に示す．計測誤差あるいは固視微動によりユーザが目的のアイコン上へカーソルを移動させることが困難であると感じた場合，確定操作を行うかあるいはマウスを動かすことで微調整を行うことができる．確定操作を行った場合，インタフェースが自動で，カーソルから最も近傍にあるアイコン上へカーソルを移動させた後に確定する（図 3 中矢印 1）．マウスを動かした場合は，カーソル位置の制御を視線からマウスに切り替える．ユーザは手でマウスを動かしてアイコン上にカーソルを移動させることができる（図 3 中矢印 2A）．カーソル位置の

制御をマウスに切り替えた後に確定操作を行った場合にも，インタフェースが自動で，カーソルから最も近傍にあるアイコン上へカーソルを移動させた後に確定する（図3 中矢印2B）．

SemiAuto 方式のユーザは次に示す細かい操作の連続によって一般的な GUI 上でも効率良くドラッグ&ドロップ操作を行うことができる．

(1) おおまかなポインティング操作1：選択したいドラッグアイコンを見る．カーソルはドラッグアイコンに近づく．
(2) 手動によるカーソル位置の微調整：必要と感じた場合，マウスを手で動かして目的のドラッグアイコンへカーソルを合わせる．
(3) プレス操作：マウスボタンを押してドラッグアイコンの選択を確定する．
(4) おおまかなポインティング操作2：選択したいドロップアイコンを見る．カーソルはドロップアイコンに近づく．
(5) 手動によるカーソル位置の微調整：必要と感じた場合，マウスを手で動かして目的のドロップアイコンへカーソルを合わせる．
(6) リリース操作：マウスボタンを離してドロップアイコンの選択を確定する．

ここで「必要と感じた場合」とは，たとえば視線により移動したカーソルの近傍に複数のアイコンがあり，選択したいアイコンがカーソルから最も近傍にあることが明らかでない場合を指す．このような場合，確定操作によって複数あるアイコンのうちどのアイコンにカーソル位置が微調整されるか分かりにくいため，手でマウスを動かして目的のアイコンへカーソル位置を調整する必要がある．

提案方式はカーソルのおおまかな移動に視線を用いるため，マウス方式と比べてマウスの移動量が少なくて済む．したがって，特にマウスを動かすことに慣れていない初心者に有効であると考えられる．

3. 実　　験

提案方式の有効性を確認するため，一般的な GUI に似せた環境で提案方式と従来のマウスのみによる操作（マウス方式）の効率を比較する実験を行った．また提案方式の利用者として，マウス方式の経験者だけでなく初心者も想定して（利き腕ではない）左手を用いた実験も行った．

3.1 実験方法

タスク：実験用ウィンドウ（**図4**）に配置されたアイコンの中から，ドラッグ元グループ中の表示が反転しているアイコンをドラッグして，ドロップ先グループ中の表示が反転しているアイコンへドロップする．ドラッグ元グループ，ドロップ先グループにはそれぞれ9つのアイコンからなる．反転しているアイコンは各グループにつき1つだけである．

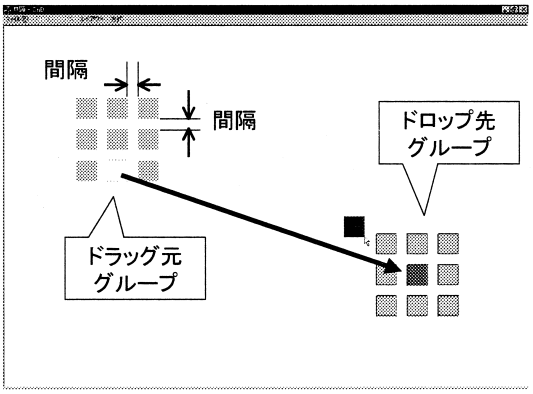


図4 実験用ウィンドウ
Fig. 4 Window for experiments.

各被験者は10回のドラッグ&ドロップ操作を連続して行う．反転するアイコンは1回の操作が成功するたびにランダムに変わる．10回のドラッグ&ドロップ操作を成功させることを1セットとし，以降で述べる実験条件（2種類のアイコンの配置間隔，2つの操作方式，実験に用いる左右の手）ごとに，5セットずつタスクを実行した．

なお試行の前には，操作方式に慣れてもらうために，操作方式ごとに10分程度の練習を行った．

被験者：日常的にマウスを用いて MS-Windows を利用している大学院生3名を被験者とした．したがって，それぞれの実験条件ごとにのべ150回のドラッグ&ドロップ操作が行われた．なお，被験者は3名とも右利きであった．

操作環境：実験には，解像度を 1,024 × 768 ピクセルに設定した21インチディスプレイを用いた．有効表示領域の大きさは縦 30 cm，横 40 cm である．顔とディスプレイとの距離が約 50 cm となるように被験者はディスプレイの前に座った．

MS-Windows のデスクトップ上のアイコンとほぼ同じ大きさになるように，アイコンの大きさは，画面上で 1.4 cm 四方の正方形とした．またアイコンの配置は，0.7 cm 間隔（狭い）と 1.4 cm 間隔（広い）の2種類を用意した．ドラッグ元グループの中心に配置されたアイコンとドロップ先グループの中心に配置されたアイコンの距離は画面上で 24.5 cm である．

視線計測：視線の計測には NAC 社製 EMR-NC を用

いた[16]．固視微動によるカーソルの振動を抑えるため，注視点座標の計測では平滑処理を行っている．具体的には，EMR-NC の時間分解能 1/30 秒において，ある時点の注視点座標は，その時点から過去 12 回の EMR-NC による座標計測値の平均値としている．

操作方式：マウスのみを使った操作方式（マウス方式）と提案方式の 2 種類を用いた．それぞれの操作方式に対して，利き手である右手と利き手でない左手を用いてタスクを実行してもらった．左手による操作は初心者による操作を想定したものである．本実験では，マウス方式に不慣れな初心者（マウスをほとんど使ったことのない人）を確保することができなかったため，次善の策として，経験者の左手によるマウス方式の操作を初心者の操作の近似とした．以降で述べるとおり，経験者の左手を用いたとしても，初心者に対する提案方式の有効性を不当に高く見積もることにはならない．

Kabbash らの分析[4]によると，アイコン近傍までカーソルを大まかに移動させる操作は，経験者の左手の方が初心者の利き手よりも効率が良い．一方，アイコン近傍からアイコンの真上へカーソルを動かす微調整の操作に限っていえば，経験者の左手と初心者の利き手はほぼ同程度の操作効率である．提案方式では，アイコン近傍までカーソルを大まかに移動させる操作には（手ではなく）視線を用いるため，経験者の左手を用いて実験したとしても，「提案方式を用いた初心者」を想定した操作時間を不当に短くすることは避けられる．

計測データ：10 回のドラッグ&ドロップ操作を 1 セットとして，1 セットに要した操作時間（秒）および試行中に発生したアイコンの選択誤り（エラー）の回数を収集した．

3.2 実 験 結 果

各操作方式ごとに右手および左手による 1 回の試行に要した時間とエラー数の平均を求めた．以降では，右手で操作を行った場合，および，左手で操作を行った場合のそれぞれについての実験結果を述べる．

右手（経験者による操作）

右手を用いた場合の 1 回の操作時間を**図 5** に，1 回の操作に対するエラーの発生回数を**図 6** に示す．1 回の操作時間の標準偏差を**表 1** に，1 回の操作に対するエラーの発生回数の標準偏差を**表 2** に示す．

アイコンの配置間隔によらずマウス方式よりも提案方式による操作時間は短かった．配置間隔が 0.7 cm の場合は 17%（0.41 秒），1.4 cm の場合は 19%（0.45 秒），提案方式の方がマウス方式よりも短かった．

エラーの発生回数については，アイコンの配置間隔によらず提案方式の方がマウス方式より多くエラーが発生した．ただし，その差は小さかった．配置間隔が 0.7 cm の場合 0.0060 回，1.4 cm の場合 0.014 回，提案方式の方がマウス方式よりも多かった．

提案方式とマウス方式の操作時間の母平均の差を検定した．検定においては，1 セットの試行に要した時間（すなわち，連続する 10 回のドラッグ&ドロップ

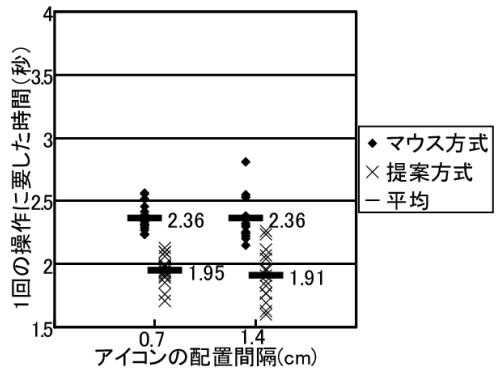


図 5 操作時間（右手：経験者）
Fig. 5 Operation time (right hand: experienced user).

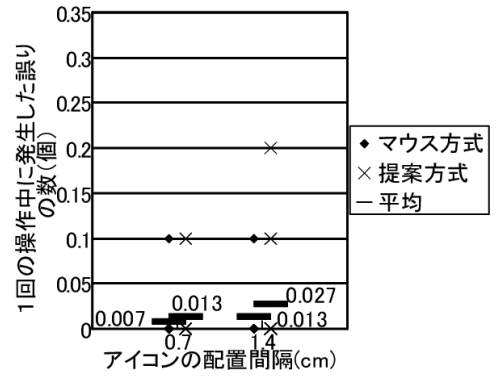


図 6 エラーの数（右手：経験者）
Fig. 6 The number of errors (right hand: experienced user).

表 1 操作時間の標準偏差（右手：経験者）
Table 1 Standard distribution of operation time (right hand: experienced user).

アイコンの配置間隔	0.7 cm	1.4 cm
マウス方式	0.11	0.18
提案方式	0.12	0.2

表 2 エラーの数の標準偏差（右手：経験者）
Table 2 Standard distribution of the number of errors (right hand: experienced user).

アイコンの配置間隔	0.7 cm	1.4 cm
マウス方式	0.026	0.035
提案方式	0.035	0.039

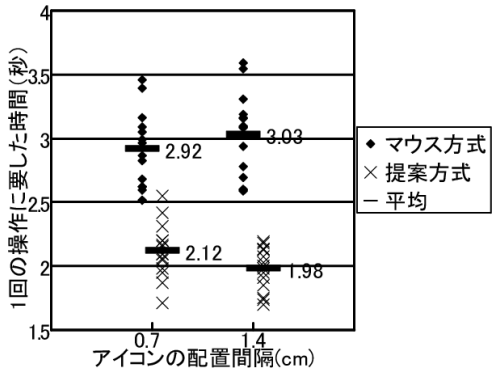


図 7 操作時間（左手：初心者）
Fig. 7 Operation time (left hand: beginner).

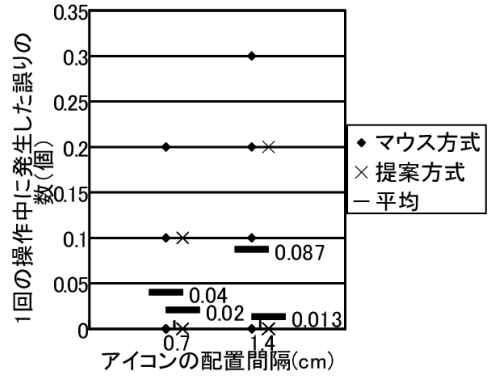


図 8 エラーの数（左手：初心者）
Fig. 8 The number of errors (left hand: beginner).

操作に要した時間）を 1 つの標本値として用い，各実験条件における 15 個の標本値について，異分散 t 検定を適用した．その結果，「提案方式による操作時間がマウス方式による操作時間よりも短い」，と有意水準 5%でいえることが分かった．エラーの回数についても，操作時間と同様に検定を行った結果，提案方式によるエラーの回数とマウス方式によるエラーの回数には有意な差はないことが分かった．

左手（初心者を想定した操作）

左手を用いた場合の 1 回の操作時間を**図 7** に，1 回の操作に対するエラー発生回数を**図 8** に示す．1 回の操作時間の標準偏差を**表 3** に，1 回の操作に対するエラーの発生回数の標準偏差を**表 4** に示す．

左手を用いた場合，アイコンの配置間隔に関係なく，マウス方式よりも提案方式の方が操作時間が短かった．配置間隔が 0.7 cm の場合 27%（0.80 秒），1.4 cm の場合 35%（1.05 秒），提案方式の方がマウス方式よりも短かった．提案方式は，右手による操作においてもマウス方式より効率が良かったが，左手による操作ではマウス方式との差がさらに顕著に表れた．実際の初心者は，熟練者の左手よりも大まかなカーソルの移動が遅いため[4]，マウス方式との差がさらに広がる可能性がある．

表 3 操作時間の標準偏差（左手：初心者）
Table 3 Standard distribution of operation time (left hand: beginner).

アイコンの配置間隔	0.7 cm	1.4 cm
マウス方式	0.29	0.32
提案方式	0.21	0.16

表 4 エラーの数の標準偏差（左手：初心者）
Table 4 Standard distribution of the number of errors (left hand: beginner).

アイコンの配置間隔	0.7 cm	1.4 cm
マウス方式	0.074	0.11
提案方式	0.041	0.052

エラーの発生回数についても，アイコンの配置間隔によらず提案方式の方がマウス方式よりも少なかった．配置間隔が 0.7 cm の場合 0.020 回，1.4 cm の場合 0.074 回，提案方式の方がマウス方式よりも少なかった．

左手による操作時間についても，右手と同様に検定を行った結果，提案方式による操作時間がマウス方式による操作時間よりも短い，と有意水準 5%でいえることが分かった．エラーの回数については，アイコンの配置間隔が 1.4 cm の場合には，提案方式によるエラーの回数は，マウス方式によるエラーの回数よりも少ないと有意水準 5%でいえることが分かった．アイコンの配置間隔が 0.7 cm 場合には，SemiAuto 方式とマウス方式には有意な差がないことが分かった．

以上の結果より提案方式は従来のマウスのみによる操作に比べて，エラーの数を大幅に増やすことなく操作時間を短縮できることが分かった．特にマウスを動かすことに慣れていない初心者を想定した場合には，操作時間が平均で 27%短縮され，エラーの数も減る可能性があることが分かった．

4. 考　　察

ターゲット選択操作の実験結果[15]より，ドラッグ&ドロップ操作の操作時間を推定し，本稿の実験結果と比較する．また，マウスのみによる操作時間との関係についても，文献 15) と比較を行う．

4.1 推定操作時間と実験結果の比較

4.1.1 定　式　化

1 回のドラッグ&ドロップ操作を，アイコンの初期知覚動作と，初期知覚後アイコンの確定に至るまでの動作に分けると，1 回の操作に要する時間 T_{DnD} は次

式で与えられる．

$$T_{DnD} = T(D_{pt1}) + T(D_{r1}) + T(D_{pt2}) + T(D_{r2})$$

ただし，

$T(D_{pt1})$：ドラッグアイコン初期知覚動作に要する時間[15)]

$T(D_{r1})$：ドラッグアイコンの初期知覚後，確定までに要する時間．文献15)における1ターゲットの知覚，認知，運動，確定の4つの動作に要する時間の合計である

$T(D_{pt2})$：ドロップアイコン初期知覚動作に要する時間

$T(D_{r2})$：ドロップアイコンの初期知覚後，確定までに要する時間

ドラッグ&ドロップ実験の1セットの試行では，10回のドラッグ&ドロップ操作を連続して行ったため，n回目のドラッグアイコン確定後ドロップアイコンを発見する（初期知覚する）までの時間 $T(D_{pt2})$ と，n回目のドロップアイコン確定後 n+1 回目のドラッグアイコンを発見するまでの時間 $T(D_{pt1})$ は，平均的にほぼ等しくなると予想される．また，ドラッグアイコンとドロップアイコンの大きさは同じであり，各アイコングループの配置間隔も同じであるため，$T(D_{r1})$ と $T(D_{r2})$ もほぼ等しくなると予想される．したがって，本稿3.1節の実験条件では，T_{DnD} は次のとおり近似される．

$$T_{DnD} \simeq 2(T(D_{pt}) + T(D_r)) \quad (1)$$

ただし，

$T(D_{pt})$：ドラッグアイコンまたはドロップアイコンの初期知覚動作に要する時間

$T(D_r)$：ドラッグアイコンまたはドロップの初期知覚後，確定までに要する時間

一方，文献15)より，1回のターゲット選択に要する時間 T_{target} は，次式で与えられる．

$$T_{target} = T(A_{pt}) + T(A_r) \quad (2)$$

ただし，

$T(A_{pt})$：ターゲット初期知覚動作に要する時間

$T(A_r)$：ターゲット初期知覚後，確定までに要する時間

4.1.2 ドラッグ&ドロップ操作とターゲット選択操作の関係

$T(D_{pt})$ と $T(A_{pt})$ の関係：

$T(D_{pt})$ は，$T(A_{pt})$ よりも少し大きな値となることが予想される．ターゲット選択実験では格子状に配置された9つのターゲットがディスプレイ上に1グループだけ存在するのに対し，ドラッグ&ドロップ実験では2グループ存在し，グループ間の距離が24.5 cm離れている．したがって，ターゲット選択操作と比較して，ドラッグ&ドロップ操作ではアイコン確定を行うたびに平均的に24.5 cmだけ余分なカーソル移動が必要となる．この移動にかかる時間を α とおくと，

$$T(D_{pt}) = T(A_{pt}) + \alpha \quad (3)$$

となる．

眼球から50 cm前方に位置する21インチディスプレイの対角上を端から端まで移動する場合に0.15 sec程度の時間を要する[7)]ことから，

$$0 < \alpha < 0.15\,\mathrm{sec} \quad (4)$$

と推定される．

$T(D_r)$ と $T(A_r)$ の関係：

2つの実験におけるターゲット確定領域の大きさが等しいときに，$T(D_r) \simeq T(A_r)$ となることが予想される．SemiAuto方式を用いた場合では，ターゲット確定領域の中で確定操作を行うとターゲットが選択される．したがって，ターゲット確定領域の大きさが，「ターゲット初期知覚後，確定までに要する時間」を左右すると考えられる．

ターゲット確定領域の大きさは，ターゲット自身の大きさとターゲットの配置間隔により決定される．ドラッグ&ドロップ実験ではアイコンの大きさが1.4 cm四方であったのに対して，ターゲット選択実験では1 cm四方であった．配置間隔についても，ドラッグ&ドロップ実験では0.7 cm，1.4 cmであったのに対して，ターゲット選択実験では0 cm，1 cm，3 cm，5 cmであった．紙面の都合上詳細な説明は割愛するが，2つの実験におけるターゲット確定領域の違いを考慮すると，$T(D_r)$ と $T(A_r)$ はおよそ次のような関係にあると推定される．

$$T(D_r)^{h=0.7} \simeq T(A_r)^{h=1.0}$$
$$T(D_r)^{h=1.4} \simeq \frac{T(A_r)^{h=1.0} + T(A_r)^{h=3.0}}{2} \quad (5)$$

$h = n$ は，ターゲットの配置間隔が n cmの場合の値を表す．

4.1.3 ドラッグ&ドロップ操作時間の推定および実験結果との比較

文献15)のターゲット選択実験の結果より，

$$T_{target} = \begin{cases} 0.83\,\mathrm{sec} & (\mathrm{h} = 1.0) \\ 0.79\,\mathrm{sec} & (\mathrm{h} = 3.0) \end{cases} \quad (6)$$

であった．

したがって，式(1)〜(6)より，

$$T_{DnD} \simeq \begin{cases} 1.66\text{〜}1.96\,\mathrm{sec} & (\mathrm{h} = 0.7) \\ 1.62\text{〜}1.92\,\mathrm{sec} & (\mathrm{h} = 1.4) \end{cases} \quad (7)$$

と推定される．これは利き腕によるドラッグ&ドロップ実験の 1 操作あたりの平均操作時間 1.95 秒（配置間隔 0.7 cm）と 1.91 秒（配置間隔 1.4 cm）に合致する．SemiAuto 方式によるドラッグ&ドロップ操作は 2 回のターゲット選択操作とほぼ同じ時間を要し，特別に大きなオーバヘッドはないといえる．

4.2 マウスのみによる操作との関係

文献 15) では，マウスのみによるターゲット選択操作の実験において，GUI 上の複数のターゲットを連続的に選択する操作（連続操作），および，ターゲット選択以外の作業（キーボード入力など）を行った後に実施されるターゲット選択操作（非連続操作）の 2 通りの状況を想定したタスクを行っている．一方，本稿の実験では，連続操作のみのタスクを行っており，ここでは連続操作についてのみ文献 15) との比較を行う．

文献 15) では，SemiAuto 方式を用いたターゲット選択操作時間は，マウスのみによる操作時間と同程度であった．一方，本稿の実験では，マウスのみによる操作と比べて，1 回のドラッグ&ドロップ操作について平均で 17%（0.4 秒）短い時間で選択できた．4.1.2 項で述べたように，ターゲット選択操作と比べると，ドラッグ&ドロップ操作ではアイコン確定を行うたびに平均的に 24.5 cm だけ余分なカーソル移動が必要となる．このカーソル移動を行うのに，マウスは視線よりも平均して 0.4 秒余分に時間を要したことが，今回の結果につながったと考えられる．

5. 関 連 研 究

従来より，主にターゲット選択を対象とした Gaze-Added Interface の研究が行われてきたが，その方法や実験条件は本稿と異なる．Salvucci らはメニューの選択操作を視線とキーボードを併用して効率良く行う方式を提案している[10]．この方式では，視線でカーソルの移動を行い，キーの打鍵により確定操作を行う．さらに，アイカメラの計測誤差や固視微動のために選びたいメニューを指し示すことができなくても，システムが操作の文脈を判断し，より選ばれる可能性の高いメニューが自動的に選択される．たとえば，デスクトップ上のアイコンを選択した直後には「ファイル」メニューをユーザが選ぶ可能性が高いとシステムが判断し，「ファイル」メニューが選択されやすくなる．ただし，選択誤りを防ぐという観点からは，本稿で対象としたアイコンのドラッグ&ドロップ操作に，Salvucci らの提案方式を適用するのは容易ではない．メニュー選択と異なり，「どの文脈においてどのアイコンが選ばれる可能性が高いか」というユーザモデルを構築するのが困難なためである．また，Salvucci らの研究では，一般的な GUI で用いられている GUI 部品よりもサイズの大きなものを用い，しかも GUI 部品の配置間隔を広くとるなどした特殊化した GUI 上で評価を行っており，本稿とは実験条件が異なる．

Zhai らの研究[17]では，ドラッグ&ドロップ操作を行う方法については言及されていないが，視線とマウスを併用してターゲット選択操作を効率良く行う方式を提案している．Zhai らの方式では，アイカメラの計測誤差や固視微動があってもターゲット選択を行うことができるよう，ユーザが手でマウスを動かしてカーソル位置を微調整できる．本稿における SemiAuto 方式は，マウスによるカーソル位置の微調整が可能である点は Zhai らの方式と同様である．ただし，カーソルがターゲット上にない場合でも，ユーザが確定操作を行うとカーソルから再近傍のターゲットが自動的に選択される点が Zhai らの方式と異なっている．また，Zhai らの研究ではターゲット間隔を 6 cm 以上あけた環境で実験を行っているが，本稿では，より一般的な GUI に近い環境で評価実験を行うために，狭い間隔（0.7 cm および 1.4 cm）でターゲットを配置した点が異なる．なお，Zhai らの論文では，ターゲット選択中に発生するエラーの数については述べられていないなめ，一般的な GUI への適用可能性が不明である．

Gaze-Centered Interface においても多くのターゲット選択操作方式が提案されているが，それらはドラッグ&ドロップ操作に応用するのには適していない．文献 1) ではターゲットを一定時間見続けることで選択できる方法が提案されている．文献 9) ではターゲットを見て眉間の筋肉を動かすことで選択できる方法が提案されている．文献 11) ではターゲットを見て瞬きすることで選択できる方法が提案されている．しかし，ドラッグ&ドロップを 2 回のターゲット選択であるとして，これらの手法を用いてドラッグ&ドロップを行うとすると，ドラッグ操作が困難になると考えられる．文献 1) の方法を用いる場合，ドラッグアイコンを見続ける必要があるため，ドロップアイコンを見ることができない．文献 9) の方法を用いる場合，ユーザはドラッグ操作を行っている間，筋肉を緊張させ続ける必要があるためユーザに負担となる恐れがある．文献 11) の方法を用いる場合には，ドラッグ操作を行っている間，目を閉じておく必要があるため，選択したいドロップアイコンを見ることができない．我々の提案方式は，マウスを併用する必要はあるが，Gaze-Centered Interface では困難なドラッグ&ドロップ操作を可能とした．

6. ま と め

一般的なGUI上で頻繁に行われるドラッグ&ドロップ操作を視線とマウスを併用して効率良く行う方式を提案した．提案方式は，ドラッグ&ドロップ操作をドラッグアイコンの選択とドロップアイコンの選択の2回の連続したターゲット選択操作であると見なし，すでにターゲット選択方式として提案されているSemi-Auto方式[15]を適用した．SemiAuto方式によりカーソル位置の微調整を行うことで，小さなアイコンが狭い間隔で配置されているような一般的なGUI上においても，ドラッグ&ドロップ操作を正確にかつ効率良く行うことができる．

一般的なGUIを想定した環境で提案方式の評価実験を行った．その結果，提案方式による操作は従来のマウスのみによる操作に比べて，エラーの数を大幅に増やすことなく，1回のドラッグ&ドロップ操作について平均で約17%（0.4秒）短縮できることが分かった．また，マウスの利用初心者を想定して左手を用いた実験では，操作誤りも減り，平均で約27%（0.8秒）短縮できることが分かった．

今後の主な課題は3つある．

(1) 提案方式の評価をさらに進めること．本稿における評価実験では，ドラッグアイコン群をデスクトップ左上に設置し，ドロップアイコン群をそこから右下方向へ平均24.5 cm離れた位置に設置して実験を行った．アイコンの方向および配置距離が異なる場合における評価を行うことが今後の課題となる．

(2) カーソルの提示方式を検討すること．評価実験では，ユーザの視線上（画面の注視点）にカーソルをつねに表示した．しかし通常のアプリケーションの利用において，カーソルをつねに表示するとカーソルが作業を妨害する可能性がある．たとえば，文献17)では，ターゲット選択操作について，ユーザが手でマウスを動かしたときに初めてカーソルを出現させることでこの問題を回避している．別の解決策としては，ユーザがアイコン近傍を見ている場合にのみカーソルを表示する方式が考えられる．SemiAuto方式にこれらの方式を取り入れることは可能である．評価実験などを通じ，操作性の高いカーソル提示方式を検討する必要がある．

(3) 文献15)で提案されているターゲット選択方式とともに，本稿で提案したドラッグ&ドロップ方式をMS-Windowsなどの現実のデスクトップに実際に適用し，操作効率の良いデスクトップを実現すること．さらに将来的には，視線によるウィンドウの選択[8]やウィンドウのスクロール[13]などの他のGUI操作方式を，一般的なGUIに適用していくことが望まれる．

謝辞 実験に協力いただいた被験者の方々に感謝する．なお，本研究の一部は，新エネルギー・産業技術総合開発機構新規産業創造型提案公募事業の援助によるものである．

参 考 文 献

1) Hansen, J., Allan, W. and Peter, R.: Eye-gaze control of multimedia systems, *Symbosis of Human and Artifact*, Anzai, Y., Ogawa, K. and Mori, H. (Eds.), Vol.20A, pp.37–42, Elsevier Science (1995).
2) 池田光男：視覚の心理物理学，森北出版 (1975).
3) Jacob, R.: What you look at is what you get: Eye movement-based interaction techniques, *Proc. ACM CHI'90 Human Factors in Computing System Conference*, pp.11–18 (1990).
4) Kabbash, P., MacKenzie, I.S. and Buxton, W.: Human performance using computer input devices in the preferred and non-preferred hands, *Proc. INTERCHI'93*, pp.474–481, ACM Press (1993).
5) 神代知範，大和正武，門田暁人，松本健一，井上克郎：視線とマウスの併用によるドラッグ&ドロップ方式の実験的評価，電子情報通信学会技術研究報告，Vol.HIP99–81, pp.37–44 (2000).
6) 久野悦章，八木 透，藤井一幸，古賀一男，内川嘉樹：EOGを用いた視線入力インタフェースの開発，情報処理学会論文誌，Vol.39, No.5, pp.1455–1462 (1998).
7) 大野健彦：視線を用いた高速なメニュー選択作業，情報処理学会論文誌，Vol.40, No.2, pp.602–612 (1999).
8) 大野健彦：視線を利用したウィンドウ操作環境，電子情報通信学会技術研究報告，Vol.HIP99–29, pp.17–24 (1999).
9) Partala, T., Aula, A. and Surakka, V.: Combined voluntary gaze direction and facial muscle acitivity as a new pointing technique, *Proc. INTERACT'01*, pp.100–107, IOS Press (2001).
10) Salvucci, D.D. and Anderson, J.: Intelligent Gaze-added interface, *Proc. CHI'2000*, pp.273–280, ACM Press (2000).
11) Shaw, R., Crisman, E., Loomis, A. and Laszewski, Z.: Eye wink control interface: Using the computer to provide the severely disabled with increased flexibility and comfort, *3rd Annual IEEE Symposium on Computer-Based Medical Systems*, pp.105–111 (1990).
12) 田崎京二，大山 正，樋渡涓二（編）：視覚情報処理，朝倉書店 (1979).
13) 大和正武，門田暁人，高田義広，松本健一，鳥居

宏次：視線によるテキストウィンドウの自動スクロール，情報処理学会論文誌，Vol.40, No.2, pp.613–622 (1999).
14) Yamato, M.: Gaze added interface suitable for General GUIs, *Proc. INTERACT '01*, pp.665–668, IOS Press (2001).
15) 大和正武，門田暁人，松本健一，井上克郎，鳥居宏次：一般的なGUIに適した視線・マウス併用型ターゲット選択方式，情報処理学会論文誌，Vol.42, No.6, pp.1320–1329 (2001).
16) 吉川 厚，大野健彦：視線を読む—ユーザにやさしい視線測定環境，*NTT R & D*, Vol.48, No.4, pp.399–408 (1999).
17) Zhai, S., Morimoto, C. and Ihde, S.: Manual and Gaze Input Cascaded (MAGIC) Pointing, *Proc. CHI '99*, pp.246–253, ACM Press (1999).

(平成13年11月26日受付)
(平成14年10月 7 日採録)



大和 正武

平成9年立命館大学理工学部情報工学科卒業．平成14年奈良先端科学技術大学院大学博士後期課程修了．同年レッドハット株式会社入社．博士（工学）．ユーザインタフェース，特に視線インタフェースに興味を持つ．電子情報通信学会，IEEE各会員．



神代 知範

平成12年奈良先端科学技術大学院大学博士前期課程修了．同年TIS株式会社入社．現在に至る．



門田 暁人（正会員）

平成6年名古屋大学工学部電気学科卒業．平成10年奈良先端科学技術大学院大学博士後期課程修了．同年同大学情報科学研究科助手．博士（工学）．ソフトウェアの知的財産権の保護，ソフトウェアメトリクス，ヒューマンインタフェース等の研究に従事．電子情報通信学会，日本ソフトウェア科学会，IEEE，ACM各会員．



松本 健一（正会員）

昭和60年大阪大学基礎工学部情報工学科卒業．平成元年同大学大学院博士課程中退．同年同大学基礎工学部情報工学科助手．平成5年奈良先端科学技術大学院大学助教授．平成13年奈良先端科学技術大学院大学教授．工学博士．Computer-Aided Empirical Software Engineering（CAESE）環境，ソフトウェアメトリクス，ソフトウェアプロセス，視線インタフェースに関する研究に従事．電子情報通信学会，IEEE，ACM各会員．