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

大和正武[†] 門田暁人[†] 松本健一[†]
井上克郎[†††] 鳥居宏次[††]

本稿では，一般的なGUI上でのターゲットのポインティング操作（ターゲットへマウスカーソルを移動し指し示す操作）に視線を利用することを目的として，ユーザの目の固視微動と視線の計測誤差の発生を考慮した3つのターゲット選択方式（Auto方式，Manual方式，SemiAuto方式）を比較検討する．(1) Auto方式では，ターゲットのサイズを仮想的に拡大する．(2) Manual方式では，ユーザが視線によるおおまかなポインティングを行った後で，ポインティング操作用デバイスをマウスに切り替える．(3) SemiAuto方式は，Auto方式とManual方式を組み合わせた方式である．一般的なGUIを想定した環境で評価実験を行った結果，SemiAuto方式による操作は従来のマウスのみを用いた操作に比べて，選択誤りを大幅に増やすことなく，操作時間は同程度かより短くなることが分かった．特に，非連続操作（カーソルの初期位置が不定の場合の選択操作）においては，操作時間を約2/3に短縮できた．

Eye Gaze and Mouse Combined Target Selection Methods for General GUIs

Masatake Yamato,[†] Akito Monden,[†] Ken-ichi Matsumoto,[†]
Katsuro Inoue[†††] and Koji Torii[††]

The purpose of this paper is to increase the efficiency of pointing operation — an operation that moves and points a mouse cursor onto a target item. We examined three pointing methods (Auto method, Manual method, and SemiAuto method) under the general GUI environment. By using the three examined methods, the computer user can select a target even if jittery motions of user's eye and the measurement error of an eye-tracking device occurred. (1) Auto method enlarges the target virtually. (2) Manual method switches the input device from the eye to the mouse after the user roughly pointed the target. (3) SemiAuto method is a method that combined Auto method and Manual method. The result of an experiment to evaluate three methods showed that the efficiency of operation with SemiAuto method is same as or faster than the mouse only operation without increasing errors largely. Especially, in the discontinuous selection situation (a target selection whose cursor position is unpredictable), SemiAuto method needed only about 2/3 of time of the mouse only operation.

1. はじめに

人間（計算機のユーザ）の視線を計算機への入力手段として用いるいわゆる「視線インタフェース」の研究がさかんに行われている．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[5]．また，Graphical User Interface（GUI）上の操作では，ユーザはまず操作対象となるGUI部品（メニュー項目やGUIボタンなど．以降ではターゲットと呼ぶ）に視線を向ける場合が多いことも知られている[14]．ポインティング操作（ターゲットへマウスカーソルを移動し指し示す操作）に限っていえば，マウスよりも視線の方が正確に効率良く，しかも自然に行える可能性が高い．したがって，ターゲットの選択操作を「ポインティング操作」と「確定操作」に分け，前者を視線で，後者をマウスボタンのクリックで行えば，GUI上のターゲット選択操作の正確さ，効率，自然さを向上できる可能性がある．

ただし，一般的なGUI上（Microsoft Windows,

† 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
†† 奈良先端科学技術大学院大学
Nara Institute of Science and Technology
††† 大阪大学大学院基礎工学研究科
Graduate School of Engineering Science, Osaka University

Mac OS，X-Window などが提供する GUI 上）でポインティング操作を視線で行う場合，ユーザの目の固視微動と視線計測装置（アイカメラ）の計測誤差が問題となる．固視微動とは，「ユーザは 1 点を見つめているつもりでも実際の視線は絶えず動いている」という現象である[2]．眼球から 50 cm 前方に位置する計算機ディスプレイ上では，固視微動によりユーザの注目する箇所（注視点）がおよそ 0.4 cm 四方の範囲で細かく振動し，それに計測誤差がおよそ 0.9〜1.7 cm 加わる[5),8),10)]．一般的な GUI 上では，ターゲットの幅（サイズ）は小さいものでおよそ 1 cm 四方となる．固視微動と計測誤差が発生する状況下で，ターゲットへマウスカーソルを正確に移動し，確定操作（ターゲットを選択するためにマウスボタンをクリックする操作）の間ターゲットを指し示し続ける，という操作を視線で行うことは容易ではない．ターゲットが接近して配置されている場合には，誤選択となる可能性も高い．

ポインティング操作を視線で行う従来の研究では，固視微動や計測誤差がユーザ操作の妨げとならないよう，比較的大きな GUI ボタンを，間隔を広くとって配置した特別な GUI を用いる場合が多い[4),5)]．また，マウスと視線で作業効率や誤り率を比較する実験も，そうした特別な GUI 上で行われる場合が多い[6),14)]．一般的な GUI 上での視線の利用を考えるのであれば，従来の提案方式や評価実験の結果をそのまま利用することは適当とはいえない．

本稿では，一般的な GUI 上でのポインティング操作に視線を利用する目的で，固視微動や計測誤差が発生してもターゲット選択を正確に効率良く視線で行うことのできる，具体的な視線・マウス併用方式を検討し，実際の GUI 環境に近い状態で評価する．ここで，一般的な GUI とは，

(C1) ポインティング操作のターゲットの幅（サイズ）が 1 cm 程度．

(C2) 隣り合うターゲット間の距離が 0 cm 以上（ターゲットが隣接している場合も想定する）．

とする．なお，以降では便宜的に，「ポインティング操作」を視線のみで行い，確定操作をマウスボタンのみで行う」といった，視線とマウスの比較的単純な組合せでターゲットを選択する方式を「基本併用方式」と呼ぶ．一方，本稿で比較検討する方式（固視微動や計測誤差に対する対策を講じている方式）を「改良併用方式」と呼ぶ．改良併用方式は次の 3 つである．

- Auto 方式：各ターゲットの周囲に確定操作を行うための領域（確定領域）を設けておき，この領域内で確定操作が行われた場合にカーソルをターゲット上へ自動的にジャンプさせる．ターゲットのサイズが仮想的に大きくなり，ユーザはターゲットを選択しやすくなる．
- Manual 方式：視線によってマウスカーソルをターゲットの近傍まで移動した（粗い位置決めを行った）後，ポインティング操作用デバイスを視線からマウスに切り替える．視線によるマウスカーソルの高速移動という特性を生かしたまま，ターゲットへの正確なカーソル移動を容易にする．
- SemiAuto 方式：上記 2 つの方式を併用する．これにより，ターゲットの配置に応じたより柔軟なポインティングを可能にする．

なお，Auto 方式と SemiAuto 方式は本稿で新たに提案する方式である．一方，Manual 方式は，文献 14) で提案されている方式とほぼ同じである．ただし，文献 14) では，(C1)，(C2) を満たす一般的な GUI 環境での評価は行われていない．

評価実験では，(C1) と (C2) を満たす実験用 GUI を準備し，これら 3 つの方式を従来方式（マウスのみによるターゲット選択）と比較する．比較の観点は，ターゲット選択操作における操作時間と誤り率である．

以降 2 章では，方式の比較検討の準備として，Fitts の法則に基づいてターゲットの選択操作を基本動作に分解し，マウスのみによる操作，および，基本併用方式による操作のそれぞれをモデル化する．3 章では，2 章での準備に基づいて，3 つの改良併用方式を比較検討し，それぞれの選択操作の流れを述べる．4 章では，5 名の被験者を対象に実施した評価実験の方法とその結果を示す．5 章で関連研究を紹介し，6 章でまとめる．

2. ターゲット選択操作モデル

2.1 マウスによる選択操作

ターゲット選択操作を，ポインティング操作 A_M と確定操作 A_S に分ける．このうちポインティング操作 A_M は，Fitts の法則[1),7)]においては，知覚動作 A_p（ターゲットとカーソルの知覚），認知動作 A_c（ターゲットとカーソルとの誤差の認知），運動動作 A_m（カーソルのターゲットへの移動）の一連の動作を最小単位（1 サイクル）とした繰返しの操作であるととらえられる（**図 1** 参照）．ターゲットのサイズ（幅）を S，i サイクル目におけるカーソルとターゲット間の距離を X_i とおくと，認知動作 A_c において $X_i < S/2$ ならばユーザは確定操作 A_S を行う．

Fitts の法則では，図 1 の一連の操作に要する時間 T_{mouse} は次の式で与えられる．

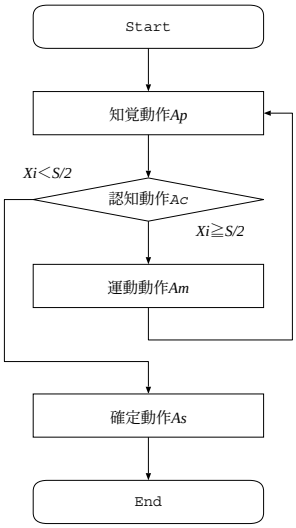


図 1 マウスによるターゲット選択の流れ
Fig. 1 Flowchart of target selection by mouse.

$$T_{mouse} = I_{mouse} \cdot \log_2 \frac{2D}{S} + T(A_S) \qquad (1)$$

$$I_{mouse} = -\frac{\tau(A_p) + \tau(A_c) + \tau(A_m)}{\log_2 \varepsilon}$$

ただし，

D：初期状態におけるカーソルとターゲット間の距離

ε：1 サイクルの動作 $(A_p \rightarrow A_c \rightarrow A_m)$ によるターゲットの接近の精度

τ：1 サイクルにおける所要時間

T：1 選択操作における所要時間

2.2 基本併用方式による選択操作

基本併用方式によるターゲットの選択操作は，主に次の 3 点においてマウスのみによる選択操作と異なる：

- GUI 上のカーソル位置にかかわらず，ターゲットに視線を移した時点で，ターゲット付近までカーソルを一瞬にして移動させることが可能である．
- 眼球には固視微動があるため，カーソルを一定箇所に停止させておくことが困難である[5),8)]．
- 注視点の座標には計測誤差が含まれるため，意図したとおりの箇所に正確にカーソルを移動させることが困難である[11),12)]．

基本併用方式によるターゲット選択操作の流れは，**図 2** のフローチャートで表される．マウスによる選択操作との違いは,「ターゲット初期知覚動作（A_{pt}）」が加わっている点である．基本併用方式による選択操作では，まず，ユーザが最初にターゲットに注目した（A_{pt}）時点で，ターゲット付近（注目箇所）へカーソルがジャンプする．次に，マウスのみによるターゲット選択と同様に，知覚動作（A_p），認知動作（A_c），運動動作（A_m）が繰り返される．そして，カーソルがターゲットの領域に入ったとユーザが判断した場合に，マウスボタンをクリックすることで確定動作（A_S）が行われる．

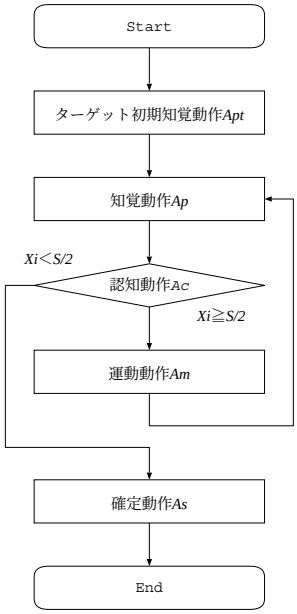


図 2 基本併用方式によるターゲット選択の流れ
Fig. 2 Flowchart of target selection by basic combination method.

ここで，固視微動の幅の平均値を e_j，アイカメラの計測誤差の平均値を e_m とおくと，固視微動と計測誤差の合計値（$e_j + e_m$）がターゲットのサイズより小さい場合（$S > e_j + e_m$）には，動作 A_{pt} によりカーソルがターゲット上にくるため，知覚動作（A_p），認知動作（A_c）の後,（$A_m \rightarrow A_p \rightarrow A_c$ のループに入らずに）確定動作（A_S）が行われる．一方，$S \leq e_j + e_m$ となる場合にはループに入るが，カーソルをターゲットに正確に近付けていくことは困難であるため，ループを回る回数はマウスと比べて非常に大きくなる．

基本併用方式によるターゲット選択に要する時間 T_{eye} は，次の式で与えられる．

- $S > e_j + e_m$ の場合：

$$T_{eye} = T(A_{pt}) + \tau(A_p) + \tau(A_c) + T(A_S) \qquad (2)$$

- $S \leq e_j + e_m$ の場合：

$$T_{eye} = T(A_{pt}) + I_{eye} \cdot \log_2 \frac{2D_{eye}}{S} + T(A_S) \qquad (3)$$

$$I_{eye} = -\frac{\tau(A_p) + \tau(A_c) + \tau(A_m)}{\log_2 \varepsilon}$$

ただし，

D_{eye}：ターゲット初期知覚動作後のカーソルとターゲットの距離 ($D_{eye} \approx e_j + e_m$)

式 (2)，(3) において，$T(A_{pt})$ の値は一般にごく小さい．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[5]．したがって，$S > e_j + e_m$ の場合には $T_{eye} < T_{mouse}$ となる可能性が高い．一方，$S \leq e_j + e_m$ の場合には $T_{eye} > T_{mouse}$ となる可能性が高い．固視微動と計測誤差のためにターゲット接近の精度 ε の値が著しく小さな値になると予想されるためである．

一般的な GUI では，式 (2)，(3) 中の定数はおおよそ次の値をとる[5),8)]．

- ターゲットのサイズ（幅）S は 1 cm 程度である．
- 眼球から 50 cm 前方に位置するディスプレイ上では，
  - 固視微動の平均値 e_j は 0.4 cm 程度
  - 注視点の計測誤差 e_m は 0.9〜1.7 cm 程度

  となる[5]．

したがって，一般的な GUI において $S \leq e_j + e_m$ となるため $T_{eye} < T_{mouse}$ となる可能性が高い．つまり，「ポインティング操作に視線を用い，確定操作にマウスボタンを用いる」という基本併用方式では，マウスのみを用いたターゲット選択よりも効率が悪くなる可能性が高い．

3. 改良併用方式

3.1 Auto 方式

$S > e_j + e_m$ とする 1 つの方法は，S を仮想的に大きくすることである．Auto 方式では，各ターゲットの周囲に確定操作を行うための領域（確定領域）を設けておき，この領域内で確定操作が行われた場合にカーソルをターゲット上へ自動的にジャンプさせる．任意のターゲット i から最も近傍のターゲットまでの距離を d_i とすると，ターゲット i の確定領域のサイズ（ターゲット i の仮想的なサイズ）は $S + d_i$ となる（**図 3**）．

Auto 方式におけるユーザの操作の流れを**図 4** に示す．ユーザは，認知動作（A_c）においてカーソルが確定領域に入った（$X_i < (S + d_i)/2$ となった）と判断した時点で，確定動作に移ることができる．なお，$X_i < (S + d_i)/2$ となった時点で自動的にターゲットの表示を変化させることで，ユーザはカーソルが確定領域に入ったことを容易に知ることができる．なお，図中には明記していないが，例外処理として，カーソルがどの確定領域にも属さない状態で確定操作を行った場合には，カーソルに最も近いターゲットの選択が確定する．

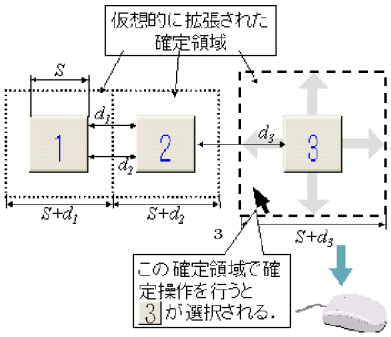


図 3 Auto 方式
Fig. 3 Auto method.

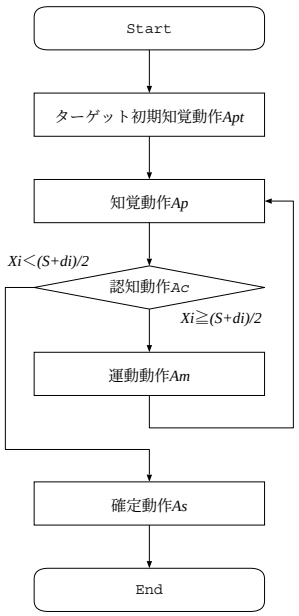


図 4 Auto 方式によるターゲット選択の流れ
Fig. 4 Flowchart of target selection by Auto method.

Auto 方式を用いることで，ターゲットのサイズが事実上大きくなり（$S \rightarrow S + d_i$），ユーザはマウスカーソルをターゲット上に移動しやすくなる．ただし，d_i が小さい場合には本方式の効果は小さい．

3.2 Manual 方式

$S > e_j + e_m$ とするもう 1 つの方法は，$e_j + e_m$ を実質的に無視できるほど小さくすることである．Manual 方式では，ユーザがマウスを動かした時点でポインティング操作用デバイスがアイカメラ（視線）からマ

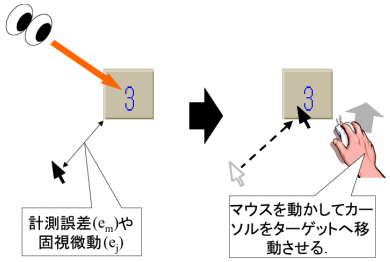


図 5 Manual 方式
Fig. 5 Manual method.

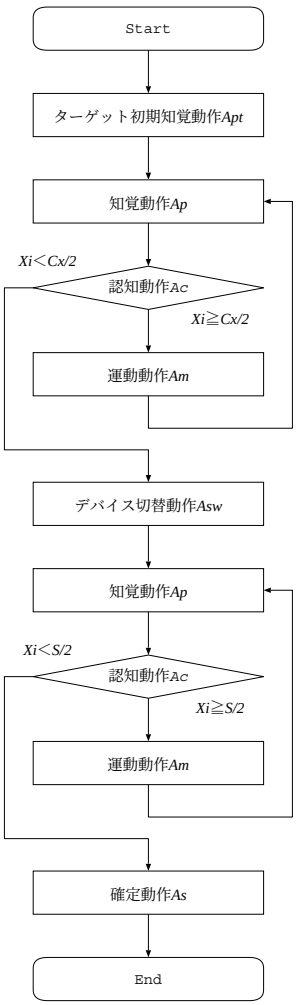


図 6 Manual 方式によるターゲット選択の流れ
Fig. 6 Flowchart of target selection by Manual method.

ウスに切り替わる．ポインティング操作用デバイスがマウスに切り替わった時点で実質的に $e_j + e_m = 0$ となり，ユーザは手でマウスを動かしてカーソルをターゲット上へ移動できる（**図 5**）．

Manual 方式におけるユーザの操作の流れを**図 6** に示す．フローチャートの前半では，ユーザは視線によるポインティング操作を行う．ユーザは，認知動作 A_c において，カーソルがターゲットに接近した（$X_i < C$ となった：C は距離を表す定数）と判断した時点で，マウスを動してデバイスの切替えを行う（動作 A_{sw}）．この時点から，ユーザの操作はフローチャートの後半に移り，マウスを用いたポインティング操作，および，確定操作を行う．なお，ユーザが（マウスをクリックして）確定操作を行った時点で，ポインティング操作用デバイスは再びアイカメラに戻る．

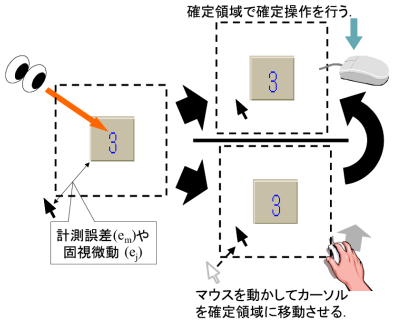


図 7 SemiAuto 方式
Fig. 7 SemiAuto method.

Manual 方式では，視線によるマウスカーソルの高速移動という特性を生かしたまま，ターゲットへの正確なカーソル移動が容易になる．

3.3 SemiAuto 方式

SemiAuto 方式は Auto 方式と Manual 方式を組み合わせた方式である（**図 7**）．Auto 方式と同様に，各ターゲットの周囲に確定領域ができる．また，Manual 方式と同様に，ユーザがマウスを動かした時点でポインティング操作用デバイスがアイカメラからマウスに切り替わる．すなわち，S を仮想的に大きくすると同時に，$e_j + e_m$ を実質的に無視できるほど小さくすることになる．

SemiAuto 方式におけるユーザの操作の流れを**図 8** に示す．ユーザは，視線によるポインティング操作をまず試みる．カーソルが仮想的なターゲットの領域に入ったと判断した（認知動作 A_{c1}）ならば，選択動作に移ることができる．また，カーソルがターゲットに接近したと判断した（認知動作 A_{c2}）ならば，マウスを動してデバイスの切替え（A_{sw}）を行うこともできる．デバイスがマウスに切り替わった後も各ターゲットの確定領域は有効であり，ユーザはマウスを手で動かしてカーソルを確定領域へ移動させることで確定操作に移ることができる．

SemiAuto 方式は，ターゲットの仮想的なサイズを

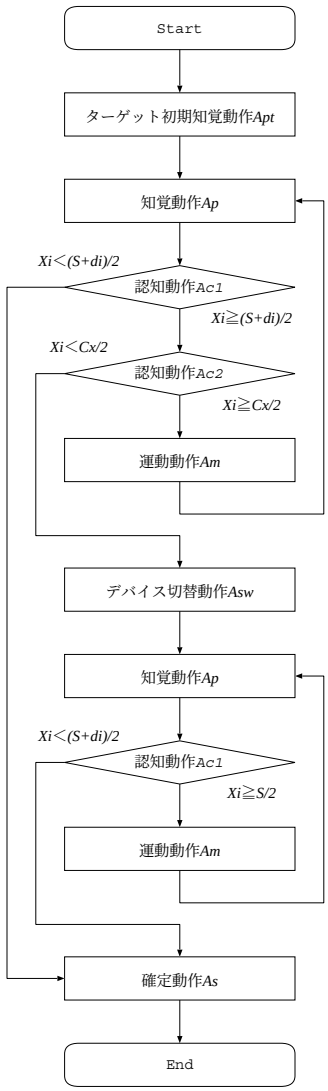


図 8 SemiAuto 方式によるターゲット選択の流れ
Fig. 8 Flowchart of target selection by SemiAuto method.

大きくすることでポインティングを容易にするという Auto 方式の利点と，マウスへのデバイス切替えを可能にすることで視線の計測誤差と固視微動を実質的にゼロにできるという Manual 方式の利点の，双方の利点を持つことになる．

4. 評 価 実 験

評価実験の目的は，3 つの改良併用方式の有効性を一般的な GUI 上で評価することである．

4.1 実 験 方 法

操作方式

マウスのみを用いた操作方式（マウス方式）と，3 つの改良併用方式（Auto 方式，Manual 方式，SemiAuto 方式）を用いた．

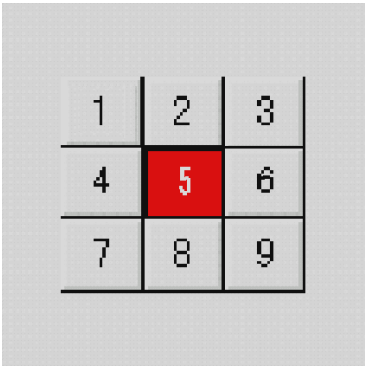


図 9 実験用ターゲット
Fig. 9 Targets for the experiments (distance between targets=0 cm).

タスク

Microsoft Windows のデスクトップ上に開かれた 1 つのウィンドウ上に，一辺の長さが 1 cm の正方形のターゲットが 9 つ配置されている．この 9 つのターゲットのなかから反転表示されているターゲットを選択する（**図 9**）．反転表示されたターゲットはつねに 1 つだけ存在し，ターゲットを 1 回選択するごとに反転するターゲットはランダムに変わる．このターゲット選択を 50 回行う．被験者に対しては，「選択はなるべく速くかつ正確に行うように」との指示が実験前に与えられている．選択を誤った場合（表示が反転していないターゲットを選択した場合や，確定操作を行ってもどのターゲットも選択されない位置にカーソルがあるにもかかわらず確定操作を行った場合）には警告音を鳴らし，正しいターゲットが選べるまで選択操作を繰り返す．

なお，マウス方式に対しては，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用意した．非連続操作を想定したタスクでは，「ユーザがターゲット選択に先だってカーソルをまず探す必要がある」という状況を想定し，1 回のターゲットの選択ごとにウインドウ上のカーソル位置をランダムに再設定した．連続操作を想定したタスクでは，カーソル位置の再設定は行わなかった．なお，マウス方式以外の 3 方式（Auto 方式，Manual 方式，SemiAuto 方式）では，カーソル位置の再設定は行っていない．これらの 3 方式では，カーソルはつねに注視点の位置に設定され続けるため，

図 10 アイカメラ
Fig. 10 Eye tracker.

カーソル位置の再設定を行うかどうかは実験結果に影響しないためである．

ターゲットの大きさと配置

ターゲットの大きさは 1 cm 四方である．9 つのターゲットを縦に 3 つ，横に 3 つずつ配置した．ターゲットの間の距離に応じて 4 つの実験用ウィンドウを用意した．ターゲット間の距離は，0 cm（隣接），1 cm，3 cm，5 cm の 4 種類である．距離が 0 cm になるようターゲットを配置したウィンドウを図 9 に示す．

被験者

奈良先端科学技術大学院大学の教官と学生の計 5 名である．被験者はいずれも日常的に Microsoft Windows を使用しており，マウス操作には慣れている．

操作環境

実験には，21 インチディスプレイを用いる．ディスプレイの解像度は 1024 × 768 ピクセル，有効表示領域の大きさは縦 30 cm 横 40 cm である．被験者はディスプレイの前に座って操作を行うが，ディスプレイ画面から被験者の顔までの距離は約 50 cm である．注視点の計測には，NAC 社製の非接触方アイカメラ EMR–NC [13] を用いた（**図 10** 参照）．被験者用計算機は Dell 社製 Dimension XPS R450（CPU Pentium II 450 MHz）である．OS は MS–Windows98 である．

計測データ

タスクが完了するまでの時間（秒），およびタスク実行中に発生した選択の誤り（エラー）の回数を収集した．

手順

タスクの実行に先だって，各操作方式ごとに 5 分間程度のタスクの練習を行った．

各被験者は，5 つの操作方式（マウス（非連続選択），マウス（連続選択），Auto，Manual，SemiAuto）のそれぞれについて 4 つの実験用ウィンドウ（ターゲットの配置間隔はそれぞれ 0 cm，1 cm，3 cm，5 cm）を用いて，合計 20 回のタスクを実行した．タスクに対する被験者の慣れが実験結果に影響する恐れがあったので，次の 2 つの点に注意してタスクの実施順序を決めた．

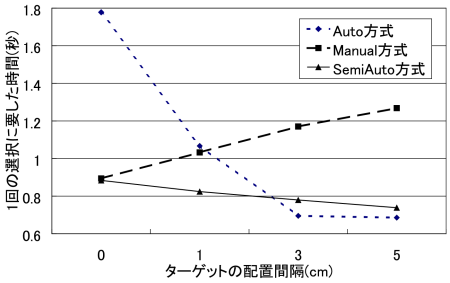


図 11 選択時間による改良併用方式間の比較
Fig. 11 Comparison between advanced combination methods in selection time.

- 実施する操作方式の順番を被験者ごとに変える．
- 各操作方式に割り当てるターゲットの配置の順序を被験者ごとに変える．

4.2 実験結果

4.2.1 改良併用方式間の比較

選択時間

ターゲットの平均選択時間とターゲット配置間隔の関係を**図 11** に示す．

Auto 方式では，ターゲットの配置間隔が広い（3 cm 以上）場合には，選択時間は小さい．しかし，配置間隔が狭くなると選択時間は大きくなった．特に，配置間隔が 0 cm の場合，Manual 方式および SemiAuto 方式による選択時間の 2 倍以上大きい（危険率 5%で有意差あり）．配置間隔 0 cm では，確定領域がターゲットのサイズと一致するため，選択が著しく困難になったと考えられる．

Manual 方式では，配置間隔が狭い（0 cm）場合には，選択時間は小さい．しかし，配置間隔が広くなるに従い大きくなった．特に，配置間隔が 5 cm のとき，Manual 方式による選択時間は Auto 方式および SemiAuto 方式による選択時間の 1.5 倍以上大きい（危険率 5%で有意差あり）．

SemiAuto 方式では，どの配置間隔でもほぼ一定の時間で選択できた．各配置間隔における選択時間は，0 cm では Manual 方式と同程度であり，1 cm では Auto 方式および Manual 方式よりも小さく，3 cm 以上では Auto 方式よりわずかに大きかった．

一般的な GUI 上ではターゲットの配置間隔が一様でないことを考えると，3 つの改良併用方式のなかでは，配置間隔にかかわらず安定して短い時間で選択できた SemiAuto 方式が有望であるといえる．

エラー数

ターゲットの平均エラー数とターゲット配置間隔の関係を**図 12** に示す．

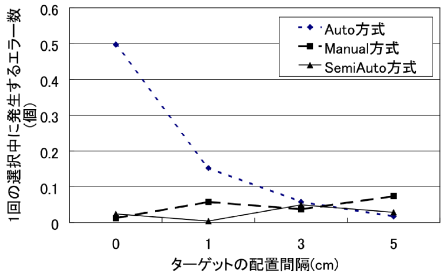


図 12 エラーの数による改良併用方式間の比較
Fig. 12 Comparison between advanced combination methods in number of errors.

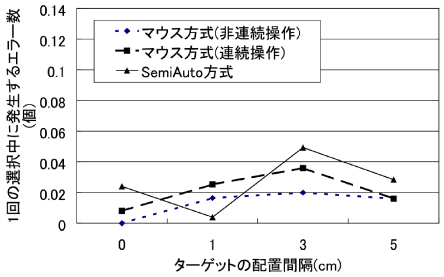


図 14 エラーの数によるマウス方式と SemiAuto 方式の比較
Fig. 14 Comparison between mouse method and SemiAuto method in number of errors.

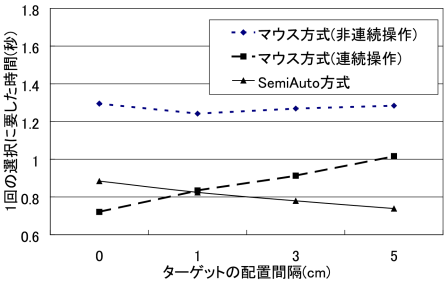


図 13 選択時間によるマウス方式と SemiAuto 方式の比較
Fig. 13 Comparison between mouse method and SemiAuto method in selection time.

Manual 方式と SemiAuto 方式の間には，大きな差はなかった（危険率 5%で有意差なし）．Auto 方式はターゲットの配置間隔が狭くなるにつれエラーの数が増加した．特に，ターゲットの配置間隔が 0 cm の場合に，Auto 方式を用いると Manual 方式および SemiAuto 方式よりも 10 倍以上多くエラーが多く発生した（危険率 5%で有意差あり）．

一般的な GUI 上では，ターゲットが隣接して配置される場合があることを考えると，Auto 方式は現実的でないといえる．

選択時間とエラーの数の結果をまとめると，3 つの改良併用方式のなかでは SemiAuto 方式がターゲットの配置間隔によらず効率良くターゲットを選択できる方式だといえる．

4.2.2 マウス方式との比較

3 つの改良併用方式のなかで，配置間隔に依らず安定して効率良く選択できた SemiAuto 方式を，マウス方式と比較する．

選択時間

ターゲットの平均選択時間とターゲット配置間隔の関係を**図 13** に示す．

非連続操作を想定した場合，GUI ボタンの配置間隔にかかわらず，SemiAuto 方式の選択時間はマウス方式の選択時間の約 3 分の 2 であった（危険率 5%で有意差あり）．被験者の行動をビデオにより分析したところ，このような結果となった理由の 1 つが，マウス方式ではユーザがマウスカーソルを画面内から探し出す必要があるのに対して，SemiAuto 方式ではその必要がないためであることが分かった．「マウスカーソルを画面内から探し出す」という動作は，ターゲット選択操作には本来不要なものであり，視線の利用がターゲット選択操作をより自然なものにした一例と考えることができる．

連続操作を想定した場合，配置間隔が 3 cm 以上の場合には，SemiAuto 方式はマウス方式より短い時間で選択できた（危険率 5%で有意差あり）．配置間隔が 1 cm 以上の場合には，SemiAuto 方式はマウス方式の選択時間は同程度であった（危険率 5%で有意差なし）．配置間隔が 0 cm の場合にのみ SemiAuto 方式による選択時間はマウス方式よる選択時間よりも約 0.2 秒大きかった（危険率 5%で有意差あり）．

配置間隔 0 cm のターゲットを連続的に選択するという特殊な状況（電卓ソフトウェアで計算を行う場合など）を除けば，SemiAuto 方式はマウス方式と同程度かより短い時間でターゲット選択を行うことができ，特に，非連続操作ではかなりの時間短縮が見込めるといえる．

エラー数

ターゲットの平均エラー数とターゲット配置間隔の関係を**図 14** に示す．

マウス方式，SemiAuto 方式のどちらもエラーの数は少なく，1 回の選択に発生するエラーの数は最大でも 0.05 回以下であった．また，ほとんどの場合において，マウス方式と SemiAuto 方式のエラー数に大きな差はなかった．連続操作を想定した場合には，すべての配置間隔において危険率 5%で有意差はなかった．

ただし，非連続操作を想定した場合には，配置間隔が0 cm および 3 cm のときのみ，SemiAuto 方式がマウス方式よりもエラー数が多かった（危険率 5%で有意差あり）．

SemiAuto 方式は，マウス方式と比較してエラーの数を大幅に増すことなくターゲット選択できるといえる．

5. 関連研究

従来よりポインティング操作を視線で行う方法が提案されている．ただしそれらのインタフェースを一般的な GUI に適用するには問題がある．

1 つ目の問題は，手でマウスなどを操作して使う通常のインタフェースに比べ，非効率的なことである．久野らは，重度肢体不自由者のコミュニケーション支援装置のインタフェースとして視線を入力に用いることを提案している[4]．久野らの提案する方式では，ユーザは 4 秒間選択対象を注視することで，選択対象の選択を確定することができる．この方式は利用者が手を使うことができないという状況では，実用的である．しかしユーザが手を使うことができる場合には，選択に確定するために時間を要するので実用的ではない．今回我々が比較検討した方式は，ユーザが手を使えることを想定して，マウスをクリックして確定操作を行うため，確定操作をより短時間で行うことができる．

2 つ目の問題は，一般的な GUI を想定して評価を行っていないことである．Sibert らは，直径 3 cm の円形をしたターゲットを 6 cm 間隔で配置した GUI を用いて，提案する視線インタフェースを用いると従来のマウスのみを用いるよりも高速にターゲットを選択できると述べている[6]．しかし，一般的な GUI では，1 cm 四方程度の大きさのターゲットが隣接して配置されている場合が少なくない．一般的な GUI 上でも Sibert らの提案するインタフェースがマウスよりも高速であるかどうかは定かではない．Zhai らの MAGIC [14]は我々の比較検討した Manual 方式と同様に視線とマウスを併用してポインティング操作を行う方式である．しかし MAGIC の評価にはターゲットを約 6 cm 以上の幅をあけて配置した GUI を用いている．また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ため，一般的な GUI への適用可能性が不明である．

6. まとめ

本稿では，一般的な GUI 上で効率良くターゲット選択操作を行う方式として，視線とマウスを併用する「Auto 方式，Manual 方式，SemiAuto 方式」の 3 つの方式を比較検討した．

一般的な GUI を想定した環境で評価実験を行った結果，SemiAuto 方式は，従来のマウスのみを用いた操作方式に比べてほぼ同程度かより効率が良いことが分かった．特に非連続操作においては，選択誤りを増すことなく，ターゲット選択時間を約 3 分の 2 に短縮できた．

従来研究においても，視線を利用したターゲット選択方式は数多く提案されてきたが，一般的な GUI 環境での使用を想定した方式はほとんどなかった．これに対し，本稿では，一般的な GUI 上におけるターゲット選択操作モデルを定義したうえで，妥当な性質を持った 3 つのターゲット選択方式（改良併用方式）を比較検討し，その有効性を実験により確認した．

今後の課題の 1 つは，カーソルの表示方式の検討である．評価実験では，ユーザの視線上（画面上の注視点）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画面の可視性が低下する可能性がある．たとえば，文献 14) では，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，ユーザがターゲット近傍を見ている場合にのみカーソルを表示する方法が考えられる．別の課題としては，提案方式を拡張してターゲット選択操作以外の GUI 操作も視線を用いて効率化することである．ターゲット選択操作以外の GUI 操作の例としてドラッグ&ドロップ操作[3]やウィンドウのスクロール[9]などがあげられる．

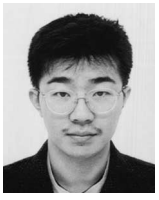
謝辞 実験にご協力いただいた被験者の方々に感謝する．なお，本研究の一部は，新エネルギー・産業技術総合開発機構新規産業創造型提案公募事業の援助によるものである．

参考文献

1) Card, S., Moran, T. and Newell, A.: *The Psychology of Human-Computer Interaction*, Lawrence Erlbaum Associates (1983).
2) 池田光男：視覚の心理物理学，森北出版 (1975).
3) 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).
4) 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).
5) 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).
6) Sibert, L.E. and Jacob, R.: Evaluation of Eye Gaze Interaction, *Proc. ACM CHI'2000 Human Factors in Computing System Conference*, pp.281–288 (2000).
7) 田村 博（編）：ヒューマンインタフェース，オーム社 (1998).
8) 田崎京二，大山 正，樋渡涓二（編）：視覚情報処理，朝倉書店 (1979).
9) 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).
10) 大和正武，神代知範，門田暁人，松本健一，井上克郎：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，Vol. ヒューマンインターフェース 84–13, pp.73–78 (1999).
11) Yamato, M., Monden, A., Matsumoto, K., Inoue, K. and Torii, K.: Button selection for general GUIs using eye and hand together, *AVI'2000 Advanced Visual Interface*, pp.270–273 (2000).
12) Yamato, M., Monden, A., Matsumoto, K., Inoue, K. and Torii, K.: Quick Button Selection with Eye Gazing for General GUI Environments, *Proc. ICS'2000 International Conference on Software: Theroy and Practice*, pp.712–719 (2000).
13) 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).
14) 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 12 年 11 月 13 日受付)
(平成 13 年 4 月 6 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．現在，奈良先端科学技術大学院大学博士後期課程に在学中．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．

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．日本ソフトウェア科学会会員．

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 年同大学教授．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．

井上 克郎（正会員）

昭和 54 年大阪大学基礎工学部情報工学科卒業．昭和 59 年同大学大学院博士課程修了．同年同大学基礎工学部助手．昭和 59 年〜61 年ハワイ大学マノア校情報工学科助教授．平成元年大阪大学基礎工学部情報工学科講師．平成 3 年同学科助教授．平成 7 年より同学科教授，現在に至る．工学博士．ソフトウェア工学の研究に従事．電子情報通信学会，日本ソフトウェア科学会，IEEE，ACM 各会員．

鳥居 宏次（正会員）

昭和 37 年大阪大学工学部通信工学科卒業．昭和 42 年同大学大学院博士課程修了．同年電気試験所（現電子技術総合研究所）入所．昭和 59 年大阪大学基礎工学部情報工学科教授．平成 4 年奈良先端科学技術大学院大学教授．現職は同大学学長．工学博士．専門はソフトウェア工学．IEEE，ACM および情報処理学会の各フェロー．