2次元ポインティングにおけるValve Iconの性能評価

木崎 駿也[1]　薄羽 大樹[1]　山田 開斗[1]　宮下 芳明[1]

概要：Valve Icon とは，オーバーシュート後にターゲットの輪郭に壁を生成し，カーソルが壁に衝突することでターゲットの上に停止し，2 回目のオーバーシュートを防ぐ手法である．先行研究では，1 次元ポインティングタスクにおいて，ターゲット幅が小さく，アイコン間の間隔が広い場合に，Valve Icon は操作時間を短縮できることが明らかとなった．しかし，実際の Graphical User Interface 上のターゲットは 2 次元である．2 次元ポインティングタスクでは，ターゲットの周辺にある非ターゲットを避けながら，ターゲットを狙うことも可能である．そのため，2 次元ポインティングにおいて Valve Icon を用いた場合，非ターゲットをオーバーシュートした場合でも，生成された壁に衝突することなくターゲットをポインティングでき，操作時間の短縮が見込める．円形のターゲットをポインティングする実験を行った結果，Valve Icon が従来手法よりも速く，正確にポインティングできる条件は存在しなかった．しかし，ターゲットの幅と高さが異なる条件で実験を行ったところ，非ターゲットが存在せず，高さが大きいターゲットであれば，Valve Icon は操作時間を短縮できた．

1. はじめに

Graphical User Interface（GUI）上のポインティングとは，カーソルを操作してターゲット（ユーザが選択したいアイコン）上にカーソルを乗せて，クリックを行うことで，ターゲットを選択する操作のことである．ターゲット上にカーソルを移動させる際に，オーバーシュート（ターゲット上をカーソルが通り過ぎること）が発生することがある．オーバーシュートが発生した場合，再びカーソルをターゲット上に乗せて，クリックを行う必要があるため，カーソルの操作時間が増加する．先行研究の Valve Icon [1] では，オーバーシュート後にターゲットの輪郭に壁を生成する．2 回目にターゲット上にカーソルが乗り，カーソルがターゲットの外に出ようとすると，カーソルが壁に衝突しターゲット上に停止するため，2 回目のオーバーシュートを防ぐことができる（図 1）．高速でターゲットをオーバーシュートし，再びターゲット方向に高速でカーソルを動かすことで，カーソルがターゲットの壁に衝突するため，より速くターゲットの選択が可能である．

先行研究 [1] では，1 次元のポインティング実験を行い，ターゲットの高さが ∞ の状況でのポインティング性能が明らかとなった．しかし，一般的な GUI のターゲットの高さは ∞ ではなく，幅と高さを持つ 2 次元ターゲットをポインティングする必要がある．また，1 次元のポインティングでは，ターゲットの両側に非ターゲット（ターゲット以外のアイコン）が存在し（ [1] の図 4 を参照），ターゲットの奥にある非ターゲットをオーバーシュートした場合，非ターゲットの壁を解除しなければターゲットを選択できない．そのため，壁の解除によって操作時間が増加する可能性がある．2 次元のポインティングでは，アイコンの周囲に空間ができるため，必ずしも非ターゲット上を通る必要はない．非ターゲットを通らなければ，非ターゲットには壁が生成されないため，壁の解除を行う必要もない．また，非ターゲットをオーバーシュートし壁が発生した場合においても，非ターゲットを避けるように操作すれば，壁を解除せずにターゲットを選択できる．そのため，1 次元のポインティングと比較して，2 次元のポインティングでは壁を解除する回数が少なくなり，Valve Icon がより有効に働く可能性がある．一方，2 次元ポインティングでターゲットの壁に衝突させるためには，ターゲット上を通るようにカーソルを動かす必要があるため，ターゲット幅が小さい状況では，1 次元のポインティングよりも困難になると考えられる．そのため，1 次元のポインティングでは，ターゲット幅が小さく，アイコン間の間隔が広い場合に，Valve Icon は有効であったが，2 次元のポインティングで

図 1　カーソルがターゲットをオーバーシュートしたとき，ターゲットの輪郭に壁が生成される．

[1]　明治大学

© 2019 Information Processing Society of Japan

は，同様の結果にならない可能性がある．本稿では，Valve Icon で 2 次元のポインティングタスクを行い，2 次元環境下で Valve Icon が有効に働く条件を調査する．

2. 関連研究

Sticky Icons [2] では，カーソルがアイコン上にある場合，ゲイン（マウスの移動量に対するカーソルの移動量）を減少させる．つまり，カーソルの速度が遅くなるため，ユーザはアイコンの幅が大きくなったような操作感が得られ，ターゲットを選択しやすくなる．Sticky Icons ではターゲットのみが存在する 1 次元タスクを行い，ゲインを少し減少させるだけでも，ポインティング時間を大幅に改善できることがわかった．しかし，非ターゲット上を通過するときにもゲインが低下するため，複数のアイコンがある条件下では，ポインティング時間が増加する可能性がある．そのため，そのような条件下での実験も行う必要があると結論づけた．

Birdlime Icon [3] は，オーバーシュートが発生する前に，カーソルの移動方向へターゲット幅を拡大し，オーバーシュートの発生を防ぐ手法である．Birdlime Icon は複数アイコンが存在する 2 次元タスクを行い，複数アイコンがある状況でもパフォーマンスが落ちることなく有効であることがわかった．

Delphian Desktop [4] では，カーソルの速度からターゲットを推測し，カーソルを推測先のターゲットに移動させる．2 次元タスクで実験を行った結果，ターゲットまでの距離が大きい場合に Delphian Desktop が有効であることがわかった．航空路マウス [5] では，マウスにジョイスティックが取りつけられており，ジョイスティックを倒した方向にあるターゲットにカーソルが移動する．複数のアイコンが存在する 2 次元タスク行った結果，航空路マウスは従来手法よりもポインティング時間が短い状況が多く見られた．Delphian Desktop や航空路マウスでは，カーソルはターゲット上で停止するため，オーバーシュートは発生しない．

ハンドジェスチャを用いたダブルクロッシング [6] は，ターゲットをカーソルが 2 回通り過ぎることによってそのターゲットを選択できる手法である．ウェブサイト上でターゲットを選択するタスクを行い，他の手法と比べ速く安定してターゲットを選択できることを示した．

Cross-drag [7] では，トリガボタンを押しながらターゲットに衝突することでターゲットをドラッグすることが可能である．非ターゲットが存在するドラッグアンドドロップのタスクを行い，多くの条件下で操作時間を短縮できた．さらに，Cross-drag を使ったウィンドウのリサイズや表計算ソフトのセルのリサイズなどのアプリケーションを実装した．

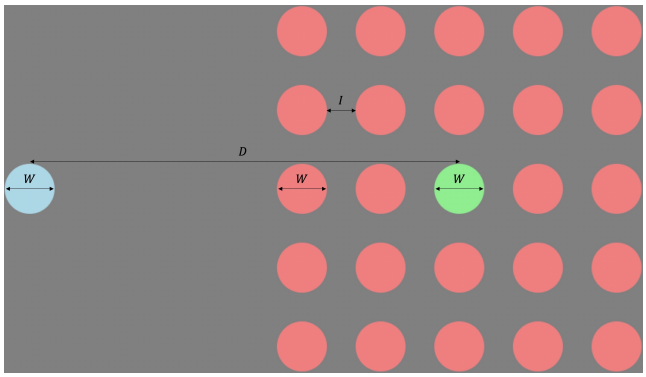

図 2 実験 1 のタスクの概要．青色：スタートエリア，赤色：非ターゲット，緑色：ターゲット．

3. 実験 1: 円形ターゲットのポインティング

3.1 実験機材

PC は MacBook Pro 16（Intel Core i9-9980H，2.3 GHz，8 cores，16GB RAM，Windows 10）を使用した．解像度は 1920 × 1200 pixels（16 inches，345.0 × 215.0 mm）であった．入力デバイスは長さ 1.80 m のケーブルに繋がれる有線マウス（Logicool G-PPD-002WL）を使用した．「ポインターの速度」は 11 段階中の 6 段階目を選択し，「ポインターの精度を高める」をオフにした．実験システムは Processing で実装されており，フルスクリーンで表示された．

3.2 実験参加者

12 名（7 名は男性，5 名は女性，平均年齢は 22.25 歳，標準偏差は 1.59 歳）が参加した．全ての参加者は右利きであり，右手でマウスを操作した．

3.3 タスク

従来手法（デフォルトのアイコン）と Valve Icon を用いて，2 次元ポインティングタスクを参加者に行わせた．PC には，青色のスタートエリア，赤色の非ターゲット，緑色のターゲットが表示された（図 2）．参加者がスタートエリアをクリックすると試行の開始を示す効果音が鳴り，計測が開始された．試行開始後，参加者はできるかぎり速く正確にターゲットを目指した．参加者がターゲット上でクリックを行った場合，その試行を成功とし，成功を示す音が鳴った．それ以外でクリックを行った場合，その試行を失敗（エラー）とし，失敗を示す音が鳴った．また，先行研究 [1] と同様，試行中のクラッチ（マウスを置き直す動作）を禁止した．参加者には，各試行の前にターゲットをどのようにポインティングするかを考えてからタスクを行うよう要求した．

2 次元ポインティングタスクでは，図 2 に示されるように，非ターゲットは 24 個配置されている．1 次元タスク

© 2019 Information Processing Society of Japan

と比べ非ターゲットが多く存在するため，どのアイコンをオーバーシュートしたかどうか識別しづらい可能性が考えられる．そのため，壁が生成されたアイコンの輪郭線を黒く表示し，壁が発生しているかどうかをユーザに提示する仕様とした．

3.4 実験デザイン

アイコンの直径（W）は，20，38，66，112 pixels（それぞれ，3.59，6.83，11.86，20.13 mm），スタートエリアの中心からとターゲットの中心までの距離（D）は，568，760 pixels（それぞれ，102.06，136.56 mm），アイコン間の間隔（I）は 0，20，38，112，∞ pixels（それぞれ，0，3.59，6.83，20.13，∞ mm）であった．比較手法（$Method$）は，従来手法（$Normal$），Valve Icon（$Valve$）の 2 種類であった．

3.5 手順

1 セットは，$4W \times 2D \times 5I = 40$ 試行であり，セット内の出現順序はランダムであった．$Method$ の順序はラテン方格法によって決められた．実験の開始前，参加者にはそれぞれのポインティング手法の説明を行い，それぞれの手法で 1 セットの練習と 12 セットの本番を行わせた．3 セットごとに 60 秒の休憩を行った．全試行は 11,520 回（$4W \times 2D \times 5I \times 2Method \times 12$ セット $\times 12$ 名）であり，1 名あたり 60 分を要した．

3.6 計測値

スタートエリアをクリックしてからターゲットをクリックするまでの操作時間 MT，エラー率を記録した．

3.7 結果

75 試行を外れ値とした [1,8]11,445 試行のうち，エラーは 602 試行であった（5.23%）．以下では，繰り返しのある分散分析で分析を行い，多重比較には Bonferroni の方法を用いた．独立変数は，$Method$，W，D，I で，従属変数は，MT，エラー率であった．グラフ中のエラーバーは標準誤差を示し，***，**，* はそれぞれ，$p < 0.001$，$p < 0.01$，$p < 0.05$ を示す．

3.7.1 学習効果

先行研究 [1] と同様に，従属変数を $Method$，セット番号（$SetNum$，1–12）として，学習効果を分析した．MT に $SetNum$ の主効果（$F_{11,121} = 4.40, p < 0.001, \eta_p^2 = 0.286$），$Method \times SetNum$（$F_{11,121} = 2.15, p < 0.05, \eta_p^2 = 0.163$）の交互作用がみられた．よって，$Method$ と $SetNum$ が MT に与える影響によって学習効果を分析する．多重比較の結果，$SetNum$ によって，それぞれの $Method$ において，MT の変化はみられなかった．そのため，全てのセットを分析の対象とした．

表 1 操作時間 MT の分散分析の結果

$Factors$	DF	$DFDen$	F	p	η_p^2
$Method$	1	11	36.6	< 0.001	0.769
W	3	33	511	< 0.001	0.979
D	1	11	134	< 0.001	0.924
I	4	44	63.1	< 0.001	0.851
$Method \times W$	3	33	19.7	< 0.001	0.641
$Method \times D$	1	11	21.9	< 0.001	0.665
$Method \times I$	4	44	18.2	< 0.001	0.623
$W \times D$	3	33	3.38	< 0.05	0.235
$W \times I$	12	132	19.3	< 0.001	0.637
$D \times I$	4	44	1.72	0.162	0.135
$Method \times W \times D$	3	33	3.92	< 0.05	0.263
$Method \times W \times I$	12	132	6.46	< 0.001	0.370
$Method \times D \times I$	4	44	1.53	0.210	0.122
$W \times D \times I$	12	132	2.96	< 0.01	0.212
$Method \times W \times D \times I$	12	132	3.12	< 0.001	0.221

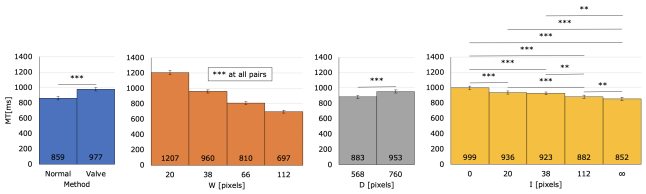

図 3 MT への影響．

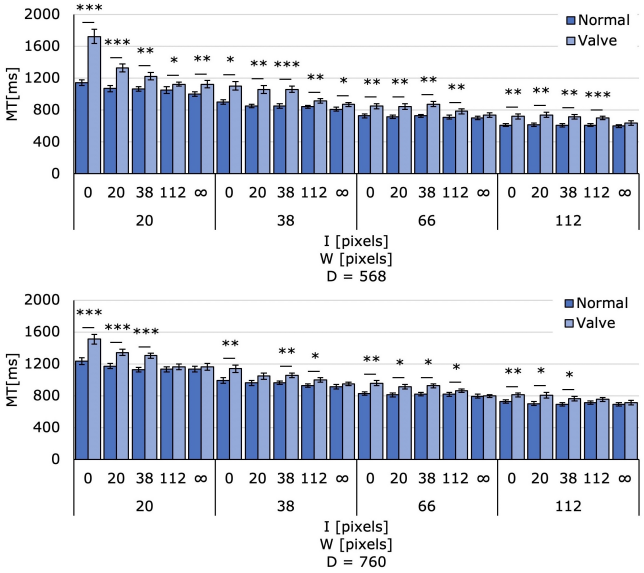

図 4 $Method \times D \times W \times I$ における MT への影響．

3.7.2 操作時間 MT

分散分析の結果は，表 1 のとおりであり，$Method$，W，D，I に主効果がみられた．$Method$，W，D，I の多重比較の結果は，図 3 の通りである．$Method \times D \times W \times I$ における操作時間への影響は図 4 の通りである．

3.7.3 エラー率

分散分析の結果は，表 2 のとおりであり，$Method$，W，D に主効果がみられた．$Method$，W，D，I の多重比較の結果は，図 5 の通りである．$Method \times W$ におけるエラー率への影響は図 6 の通りである．

© 2019 Information Processing Society of Japan

表 2　エラー率の分散分析の結果

Factors	*DF*	*DFDen*	*F*	*p*	η_p^2
Method	1	11	7.98	< 0.05	0.421
W	3	33	13.9	< 0.001	0.558
D	1	11	1.25	0.288	0.102
I	4	44	3.98	< 0.01	0.265
Method × *W*	3	33	5.02	< 0.01	0.313
Method × *D*	1	11	0.465	0.509	0.041
Method × *I*	4	44	0.953	0.443	0.080
W × *D*	3	33	1.95	0.141	0.150
W × *I*	12	132	0.836	0.613	0.071
D × *I*	4	44	0.309	0.870	0.027
Method × *W* × *D*	3	33	0.214	0.886	0.019
Method × *W* × *I*	12	132	1.34	0.203	0.109
Method × *D* × *I*	4	44	0.568	0.687	0.049
W × *D* × *I*	12	132	1.33	0.209	0.108
Method × *W* × *D* × *I*	12	132	0.509	0.906	0.044

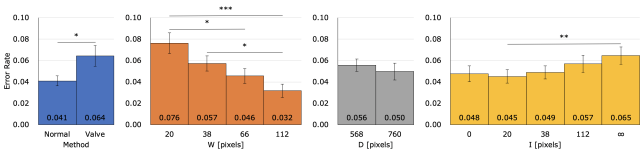

図 5　エラー率への影響.

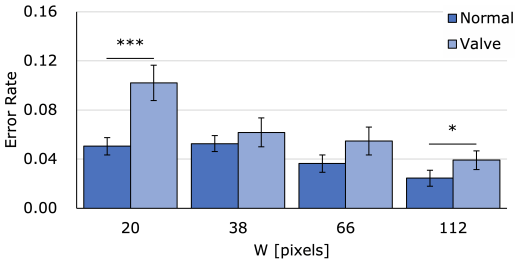

図 6　*Method* × *W* におけるエラー率への影響.

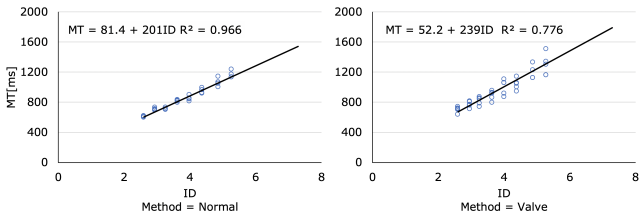

図 7　各 *Method* におけるモデル適合度.

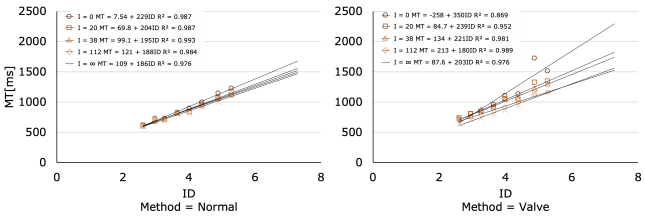

図 8　各 *I* における *Method* のモデル適合度.

3.7.4　モデル適合度

フィッツの法則 [9,10] によれば，*D* だけ離れた幅 *W* のターゲットを選択するまでの操作時間 *MT* は，2 つの回帰定数（*a* と *b*）を用いて，次のように表わせる．

$$MT = a + b\log_2\left(\frac{D}{W} + 1\right) \tag{1}$$

図 3 と図 5 で示されるように，*W* と *D* には，フィッツの法則における *W* と *D* と同様に，ターゲットの幅が大きく距離が短いほど操作時間が短い効果がみられた．そのため，フィッツの法則をポインティングの操作時間予測モデルとし，適合度を検証した．*Method* 別にモデル適合度をみると，*Normal* では $R^2 = 0.966$ であり，*Valve* では $R^2 = 0.776$ であった（図 7）．*Normal* では $R^2 > 0.900$ であるため，*Normal* はフィッツの法則に適合した [11]．*Valve* では，*I* 間に差がみられたため，*I* 別にもモデル適合度を検証した．その結果，*Normal* ではすべての *I* において $R^2 > 0.900$ であったが，*Valve* では $I = 0$ を除いて $R^2 > 0.900$ であった（図 8）．

3.8　考察

2 次元タスクでは Valve Icon は従来手法よりも遅かった（図 3）．1 次元タスクでは Valve Icon が従来手法よりも遅くなることはなかった．2 次元タスクでは，壁に衝突させるためにはターゲット上を通るようにカーソルを操作する必要があり，カーソルを高速で操作できないため，操作時間を短縮できなかったと考えられる．実際に，多くの実験参加者が $I = \infty$ のとき（非ターゲットが存在しないとき）に，意図的にターゲットをオーバーシュートし，生成される壁にカーソルを衝突させてポインティングを行っていたと述べているが，操作時間は短縮できていなかった（図 4）．

Valve Icon は $I = 0$ のときフィッツの法則に適合せず，それ以外の条件では適合した（図 8）．1 次元タスクでは $I = 0$，$I = \infty$ のときに適合しなかった．2 次元タスクでの $I = 0$ の条件下では，1 次元タスクと同様，非ターゲットをオーバーシュートしやすい条件である．そのため，非ターゲットをオーバーシュートしたときに非ターゲットに生成される壁の解除や，壁を避けながらターゲットのポインティングを行うことで，操作時間が増加し，適合しなかったと考えられる．また，2 次元タスクで $I = \infty$ のときに適合した理由として，Valve Icon が操作時間を短縮できていないためだと考えられる．

ターゲットの幅が最も狭い場合（$W = 20$），最も広い場合（$W = 112$）に Valve Icon のエラー率は高かった．1 次元タスクでは，ターゲットの上を必ず通り過ぎるため，壁に衝突させやすく，Valve Icon のエラー率が高くなることはなかった．2 次元タスクでは，ターゲットの上を通らない場合もあり，壁に衝突させることが困難なため，Valve Icon のエラー率が高くなったと考えられる．特に，ターゲットの幅が最も狭い場合（$W = 20$），2 次元タスクではターゲット上をカーソルが通り過ぎることが困難であり，壁に衝突することがなくカーソルが通り過ぎてしまう状況が見られる（図 9）．また，ターゲットの幅が大きい場合

© 2019 Information Processing Society of Japan

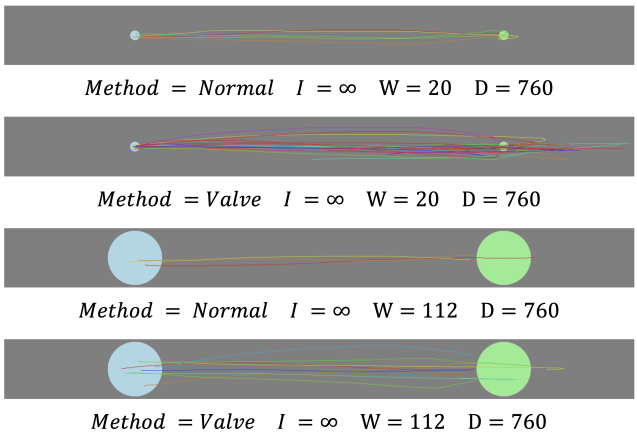

図 9 各条件でエラーとなった試行のカーソルの軌跡.

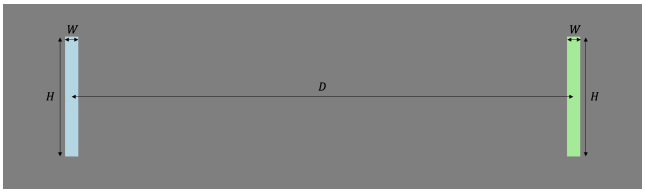

図 10 実験 2 のタスクの概要．青色：スタートエリア，緑色：ターゲット.

($W = 112$)，ターゲットの手前でクリックを行い失敗する，ターゲットの壁に衝突させようとして失敗する場面が見られた.

4. 実験 2: 幅と高さが異なるターゲットのポインティング

実験 1 では，Valve Icon が有効に働く状況は存在しなかった．一方で，1 次元タスク [1] では，ターゲットの幅が狭く，アイコン間の間隔が広い条件で，Valve Icon は速かった．1 次元タスクの条件下では，ターゲットの高さが ∞ であるため，ターゲット方向にカーソルを動かしたときに必ずターゲットを通る．そのため，オーバーシュートしやすく，また，壁にも衝突させやすく，Valve Icon は操作時間を短縮できると考えられる．2 次元タスクにおいても，ターゲットの幅が狭く，高さがある程度大きい状況では，ターゲット上を通ることが容易であるため．Valve Icon が有効に働く可能性がある．実験 2 では，従来手法と Valve Icon で幅と高さが異なる 2 次元ターゲットのポインティングを行い，2 次元環境下において，Valve Icon が有効に働くために必要な高さを確認する（図 10）．また，1 次元タスクで有効と示唆された動画プレイヤーのシークバーなどの検証のため，非ターゲットがない条件で実験を行う.

実験機材，タスクの手順，計測値は実験 1 と同様であった.

4.1 実験参加者

4 名（全てが男性，平均年齢は 21.25 歳，標準偏差は 1.50 歳）が参加した．全ての参加者は右利きであり，右手でマウスを操作した．内 2 名は実験 1 にも参加した.

表 3 操作時間 MT の分散分析の結果

$Factors$	DF	$DFDen$	F	p	η_p^2
$Method$	1	3	45.0	< 0.01	0.938
W	3	9	40.1	< 0.001	0.930
H	7	21	26.1	< 0.001	0.897
$Method \times W$	3	9	63.3	< 0.001	0.955
$Method \times H$	7	21	11.5	< 0.001	0.793
$W \times H$	21	63	0.856	0.644	0.222
$Method \times W \times H$	21	63	2.55	< 0.01	0.459

4.2 実験デザインと手順

アイコンの幅（W）は，20，38，66，112 pixels（それぞれ，3.59，6.83，11.86，20.13 mm），アイコンの高さ（H）は，20，60，100，140，180，220，260，300 pixels（それぞれ，3.59，10.78，17.97，25.16，32.34，39.53，46.72，53.91 mm），距離（D）は，760 pixels（136.56 mm），であった．比較手法（$Method$）は，従来手法（$Normal$），Valve Icon（$Valve$）の 2 種類であった.

1 セットは，$4W \times 8H = 32$ 試行であり，セット内の出現順序はランダムであった．$Method$ の順序はラテン方格法によって決められた．実験の開始前，参加者にはそれぞれのポインティング手法の説明を行い，それぞれの手法で 1 セットの練習と 15 セットの本番を行わせた．3 セットごとに 60 秒の休憩を行った．全試行は 3,840 回（$4W \times 8H \times 2Method \times 15$ セット $\times 4$ 名）であり，1 名あたり 60 分を要した.

4.3 結果

30 試行を外れ値とした 3,810 試行のうち，エラーは 251 試行であった（6.54%）．以下では，繰り返しのある分散分析で分析を行い，多重比較には Bonferroni の方法を用いた．独立変数は，$Method$，W，D，I で，従属変数は，MT，エラー率であった．グラフ中のエラーバーは標準誤差を示し，***，**，* はそれぞれ，$p < 0.001$，$p < 0.01$，$p < 0.05$ を示す.

4.3.1 学習効果

従属変数を $Method$, セット番号($SetNum$, 1–15)として，学習効果を分析した．MT（$F_{14,42} = 1.27, p = 0.265, \eta_p^2 = 0.298$）とエラー率（$F_{14,42} = 0.328, p = 0.986, \eta_p^2 = 0.0986$）において $SetNum$ の主効果がみられなかったため，全てのセットを分析の対象とした.

4.3.2 操作時間 MT

分散分析の結果は，表 3 のとおりであり，$Method$，W，H に主効果がみられた．$Method$，W，H の多重比較の結果は，図 11 の通りである.

4.3.3 エラー率

分散分析の結果は，表 4 のとおりであり，すべての要因において，主効果はみられなかった．$Method$，W，H の

© 2019 Information Processing Society of Japan

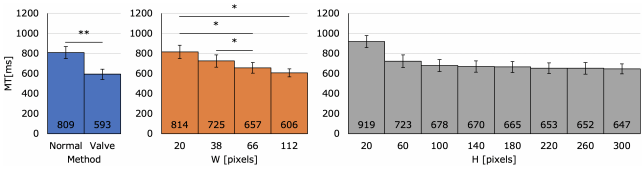

図 11 MT への影響.

表 4 エラー率の分散分析の結果

$Factors$	DF	$DFDen$	F	p	η_p^2
$Method$	1	3	2.81	0.192	0.484
W	3	9	1.05	0.416	0.260
H	7	21	1.41	0.254	0.319
$Method \times W$	3	9	3.04	0.0853	0.503
$Method \times H$	7	21	0.763	0.624	0.203
$W \times H$	21	63	0.768	0.746	0.204
$Method \times W \times H$	21	63	0.767	0.746	0.204

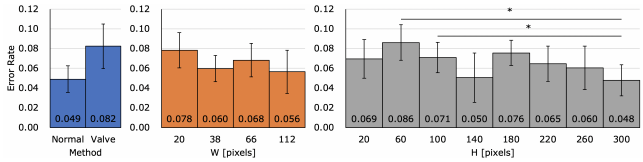

図 12 エラー率への影響.

多重比較の結果は，図 12 の通りである．

4.4 考察

$W = 20$ のとき，$H > 20$ において，Valve Icon は従来手法よりも速かった（図 13）．ターゲット幅が小さいため，オーバーシュートしやすく，高さが大きいときにはカーソルを壁に衝突させやすいため，操作時間を短縮できたと考えられる．

$W \geq 38$ のとき，$H > 60$ において，ほとんどの条件で Valve Icon は速かった．ターゲット幅が大きく，オーバーシュートが発生しづらい条件でも，オーバーシュートを恐れることなくカーソルを高速に操作できるため，操作時間を短縮できたと考えられる．

$H = 20$ のとき，Valve Icon と従来手法に差はみられなかった．これは壁に衝突させづらい条件下であり，オーバーシュートが発生した場合においても，壁に衝突させることなく直接ターゲットを目指してカーソルを動かしたため差が出なかったと考えられる．

$H = 20$ かつ $W = 20$ のときに差はみられなかった．$H = 100$ かつ $W = 112$ のときに Valve Icon は速かった．実験 1 のこれらと似た条件下では，$I = \infty$ において，$W = 20$ のときに Valve Icon は遅く，$W = 112$ のときに差はみられなかった．実験 2 では，Valve Icon が有効に働く条件が多く，ユーザがオーバーシュートを恐れずに Valve Icon での操作を行ったため，従来手法よりも遅くなることがなかったと考えられる．一方で，実験 1 では Valve Icon が有効に働く状況が少なく，ユーザが積極的に Valve Icon の壁を使用しなかったため，従来手法よりも速くなることがなかったと考えられる．そのため，例えば，実験 1 の条件においても，I を固定した順序で実験を行えば，$W = 20$ や $W = 112$ のときにも Valve Icon は有効に働くと考えられる．

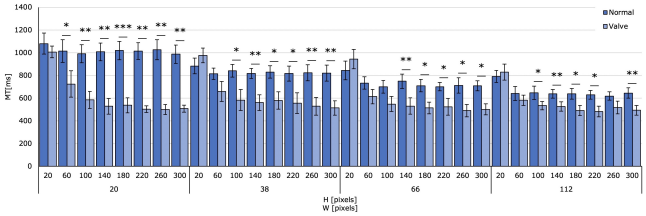

図 13 $Method \times W \times H$ における MT への影響.

Valve Icon と従来手法ではエラー率に差はみられなかった．1 次元タスクの結果から，アイコンの高さがある程度大きければ，エラー率は低下すると考えられる．ターゲットが縦長になる条件が多く，ターゲットの壁に衝突させやすいため，Valve Icon のエラー率が増加しなかったと考えられる．

以上の結果から，2 次元ポインティングにおいても，ターゲットの高さがある程度大きい場合，Valve Icon は操作時間を短縮できることがわかった．そのため，先行研究で示唆された動画プレイヤーのシークバーや，拡張ディスプレイ時のメニューバーへの Valve Icon の適用は有効であると考えられる．

5. 結論

オーバーシュート後にターゲットの輪郭に壁を生成し，カーソルを壁に衝突させてポインティングを行う「Valve Icon」で，2 次元ポインティングのタスクを行った．円形ターゲットのポインティングタスク（実験 1）の非ターゲットが存在しない条件では，Valve Icon と従来手法に操作時間の差はなかった．また，アイコン間の間隔が狭い条件や，ターゲットまでの距離が短い条件において，Valve Icon は従来手法よりも遅くなった．実験結果より，Valve Icon が有効な状況は存在しなかった．一方で，幅と高さが異なるターゲットのポインティングタスク（実験 2）では，ターゲットの高さが大きい 2 次元ターゲットであれば，Valve Icon は操作時間を短縮できることがわかった．以上の結果から，Valve Icon はターゲットが一方向に長く，ターゲットのみが存在する状況では有効であり，先行研究で有効であると示唆された「シークバーやメニューバーなどの細長い形状」に適用すべきであるといえる．

参考文献

[1] 木崎駿也，薄羽大樹，山田開斗，宮下芳明：Valve Icon: オーバーシュート後に生成される壁を用いたポインティング高速化手法の提案，研究報告ヒューマンコンピュータインタラクション (HCI)， Vol. 2019-HCI-185, No. 23, pp. 1–8 (2019).

© 2019 Information Processing Society of Japan

[2] Mandryk, R. L. and Gutwin, C.: Perceptibility and utility of sticky targets, *Proceedings of graphics interface 2008*, Canadian Information Processing Society, pp. 65–72 (2008).

[3] 築谷喬之，高嶋和毅，朝日元生，伊藤雄一，北村喜文，岸野文郎：Birdlime icon: 動的にターゲットを変形するポインティング支援手法，コンピュータ ソフトウェア，Vol. 28, No. 2, pp. 2_140–2_152 (2011).

[4] Asano, T., Sharlin, E., Kitamura, Y., Takashima, K. and Kishino, F.: Predictive interaction using the delphian desktop, *Proceedings of the 18th annual ACM symposium on User interface software and technology*, ACM, pp. 133–141 (2005).

[5] 中辻智裕，山本景子，　倉本到，辻野嘉宏：航空路マウス：親指ジョイスティックでのターゲット方向指示によるポインティング高速化手法，インタラクション 2016 論文集，pp. 980–985 (2016).

[6] 中村　卓，高橋　伸，田中二郎：大画面環境におけるハンドジェスチャの選択手法：ダブルクロッシングの提案と他の選択手法との比較 (ヒューマンコンピュータインタラクション)，電子情報通信学会論文誌. D, 情報・システム， Vol. 96, No. 4, pp. 978–988 (2013).

[7] 山中祥太，宮下芳明：細長いターゲットのドラッグ開始を支援する手法とその評価，コンピュータソフトウェア， Vol. 33, No. 1, pp. 1_111–1_125（オンライン），DOI: 10.11309/jssst.33.1_111 (2016).

[8] Soukoreff, R. W. and MacKenzie, I. S.: Towards a standard for pointing device evaluation, perspectives on 27 years of Fitts' law research in HCI, *International Journal of Human-Computer Studies*, Vol. 61, No. 6, pp. 751–789, DOI: https://doi.org/10.1016/j.ijhcs.2004.09.001 (2004).

[9] Fitts, P. M.: The information capacity of the human motor system in controlling the amplitude of movement., *Journal of experimental psychology*, Vol. 47, No. 6, p. 381 (1954).

[10] MacKenzie, I. S.: A Note on the Information-Theoretic Basis for Fitts' Law, *Journal of Motor Behavior*, Vol. 21, No. 3, pp. 323–330, DOI: 10.1080/00222895.1989.10735486 (1989).

[11] Soukoreff, R. W. and MacKenzie, I. S.: Towards a Standard for Pointing Device Evaluation, Perspectives on 27 Years of Fitts' Law Research in HCI, *Int. J. Hum.-Comput. Stud.*, Vol. 61, No. 6, pp. 751–789, DOI: 10.1016/j.ijhcs.2004.09.001 (2004).

© 2019 Information Processing Society of Japan