

# ポインティングによる多数項目からの選択インタフェースの設計と評価

越澤 勇太<sup>†</sup> 日浦 慎作<sup>†</sup> 佐藤 宏介<sup>†</sup>

記憶装置の大容量化に伴い、何らかの順に並べられた多数の項目から選択を行う機会が増えている。GUI ではそのような場合スクロール表示が用いられることが多いが、スクロールバーでは全体の中での選択対象のおおよその位置が分かりづらく、選択対象が画面内に入るまで試行錯誤が必要になることがある。またスクロールは項目が少ない場合には必要のない操作であり、項目数の大小により選択操作のメンタルモデルに相違が生じる。我々はこの問題を解決するために、マルチスケールインタフェースにより対象徐々に近づくことで多数項目からの選択を実現する。このときポインタが向かう先の項目を移動させないような拘束条件のもとで画面を拡大する。提案手法では項目が大幅に増加しても操作性の低下を抑制可能であることを被験者実験により示した。

## Design and Evaluation of Pointing Interfaces for Selecting from Multiple Targets

YUTA KOSHIZAWA,<sup>†</sup> SHINSAKU HIURA<sup>†</sup> and KOSUKE SATO<sup>†</sup>

Capacity of storage devices is increasing day by day and it causes more situations that people select one from a number of targets. For such cases, scrollbars are commonly used in GUI. However, mental models between selecting one from a small number of targets and thousands of targets are different because users do not need to scroll the targets when the number of targets is enough small. We propose novel interfaces named “Ring” and “Arc” to solve the problem and evaluate them through user tests.

### 1. 序 論

近年の記憶装置の大容量化に伴い、大量の音楽、画像、動画等のファイルを扱う機会が増えている。これらは必ずしも適切に分類されているとは限らないが、曲名順や撮影日時などによる順序づけが可能であることが多い。また逆に、順序づけも分類もされていない多数の項目から目標の項目を見つけ出すことはそもそも大変困難である。そこで本論文では、順序づけされた多数の項目から1つの項目を探し、選択するための方法について論じ、これを効率的に行うためのインタフェースを提案する。

GUI(Graphical User Interface)ではマウス等のポインティングデバイスを用いてカーソルを操作し、ポインティングによって選択操作を行う。しかし、ファイルの数が膨大になると、ディスプレイという限られた空間上に全ての項目を表示することができないため、目標の項目に対して直接ポインティングを行えない。このような場合、GUIではスクロールバーが用いられることが多い。スクロールバーはディスプレイに表

示される範囲よりも大きな領域のうち、表示される部分をスクロール操作によってコントロールするというアイデアの基に作られている。しかし、そのようなインタフェースでは目標の項目がどのあたりにあるのかが分からず、目標項目が画面内に来るまで二分探索のような操作で試行錯誤を繰り返す必要がある。また項目数が大きい場合のみ表示範囲をコントロールする必要があり、項目数の大小により操作のメンタルモデルが変化してしまう。

我々はそのような試行錯誤やメンタルモデルの相違を解消するためにスケールの変化(マルチスケール)を導入する。スケールを変化できると、スケールを小さくして全項目をディスプレイ上に表示し、クリックを行うときにはスケールを大きくして項目にポインティングしやすくすることが可能である。しかし、ユーザが意識的にスケール操作をしなければならないのであれば、スクロールバーでスクロール操作とポインティング動作の2つが必要なようにわずらわしい。通常、マルチスケールインタフェースではズームとパンの二つの操作が必要になる<sup>1)</sup>。パンの操作はカーソルのポインティングとは別の、2次元空間上での表示範囲の変更である。ディスプレイ上のカーソルのポインティングという2次元の操作に加え、パンの2次元操作、

<sup>†</sup> 大阪大学 大学院基礎工学研究科  
Graduate School of Engineering Science, Osaka University

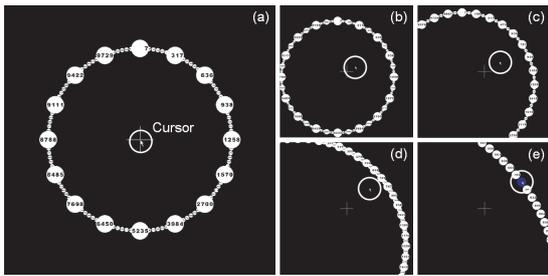


図 1 “Ring”: 提案する多数項目からの選択インタフェース

ズームの1次元操作を行うとユーザは5次元的な操作を行わなければならない。これらを単純なカーソル操作だけで行うことは難しい。そこで、我々はあえて1次元的に項目を配置することでそれぞれの操作の次元を下げ、単純なカーソルのポインティングによるマルチスケール操作を実現した。また特に、カーソルが向かう先のターゲットが画面上で移動しないような制約条件を与えることで、カーソルの軌跡を必要以上に迂回させる必要がなくなるとともに、ズーム操作を意識的に行う必要がなく、初めて触れる人でも自然なポインティング操作の一環として対象を選択出来るようなインタフェースの実現を意図している。

図 1 は我々の提案するインタフェースの一つ、“Ring”の動作を示したものである。項目を環状に配置し、カーソルを円の中心に移動すると全項目が表示されるようにズームアウトする。カーソルを中心から円周へと移動するに従ってカーソル移動方位に配置されている項目にズームインする。ただし、スケールが小さいときには項目を読み取ることができないので、いくつかの項目を拡大表示することでインデックスとし、これを参考に対象へポインタを近づける。“Ring”ではカーソルの移動によってパン、ズーム、項目へのポインティングの全てを可能にしている。また、項目数の大小に関わらず、項目の方位へ移動しポインティングするという単一の操作によって選択が可能になっている。

## 2. 関連研究

ポインティングのユーザビリティを評価するために、Fittsの法則<sup>2)3)</sup>がよく用いられる。Fittsの法則は、ポインティングにおけるカーソルの移動時間  $MT$ (Movement Time)、カーソルからターゲットまでの距離  $A$ (Amplitude)、ターゲットの幅  $W$ (Width) の関係をモデル化したもので次の式で表される。

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

ただし、 $a, b (b > 0)$  は経験的または実験的に導かれる定数である。 $a$  は認知に要する時間と考えられているため、 $\log_2 \left( \frac{A}{W} + 1 \right)$  はポインティングの難易度  $ID$ (Index of Difficulty) と呼ばれる。Fittsの法則によると、 $MT$  を小さくするためには  $A$  を小さくするか  $W$  を大きくすることが必要であることがわかる。

$A$  を小さくする手法としては、Baudisch らの提案する Drag-and-pop<sup>4)</sup> や Guiard らの提案する Object Pointing<sup>5)</sup> が挙げられる。一方、 $W$  を大きくする手法としては、McGuffin らの直接的にターゲットを拡大表示する手法<sup>6)</sup> や、Kabbash らの Area Cursor<sup>7)</sup>、それを改良した Grossman らの Bubble Cursor<sup>8)</sup> のようにカーソルに大きさを持たせることで仮想的に  $W$  を大きくする手法等が見られる。

しかし、これらの研究はディスプレイ上に配置された項目を選択する場合のパフォーマンスを向上させるためのものである。ディスプレイ上に表示しきれないほど大量の項目が存在する場合にはスクロールバーなどの、画面より広い領域の一部を画面内に導入する方法が広く用いられている。しかしスクロールバーでは、先に述べたようにターゲットが画面内に現れるまで試行錯誤が必要である。また項目が極端に多い場合、ターゲットを画面内にとどめることが難しいほどスクロールバーが敏感になることもある。後者の問題については操作部位や操作方法により敏感さを変えることが出来る手法<sup>23)~25)</sup> が提案されているが、スクロールバーの見た目を踏襲しているがゆえにむしろ、操作方法の教示と習熟が必要であると思われる。これに加えスクロールバーの脇にインデックスを付けることで前者の問題の解決も図った手法<sup>21)</sup> も提案されているが、全体を一覧するという観点では全体像から徐々に詳細の表示に移行するマルチスケールインタフェースが優れていると思われる。

マルチスケールインタフェースにおいても、スクロールバーと同様に操作の敏感さの問題が存在することが過去の研究で示されている。Furnas<sup>9)</sup> や Sarkar ら<sup>10)</sup> の提案する Fisheye View は広大なドキュメントをブラウジングするには適しているがターゲット選択においてはユーザビリティの向上が見込めない<sup>6),11),12)</sup>。なぜなら表示だけが拡大されても操作の敏感さが改善されないからであり、この問題については次節で論じる。スケールの拡大に応じて操作の敏感さを調整する手法として、OrthoZoom Scroller<sup>26)</sup> はポインタの位置でスケールと感度を調整しており、本論文が提案する手法に最も近い手法であるが、ポインタの位置とスケールの関係は直観的であるとは言えず、操作法の教

示が必要であると思われる。そこで我々はポインタとターゲットの距離によりスケールを調整することと、ポインタが向かう先のターゲットが移動しないという制約を設けることで、通常のポインティング操作との差異が小さいマルチスケールインタフェースを提案する。

操作の初期段階で、全項目を単に縮小表示する方法では目的のターゲットを視認できない。そこで目安の表示が必要となるが、LensBar<sup>22)</sup>では各項目に重要度をあらかじめ設定することで適切に間引き表示する手法を提案しており、これを本手法に導入することでより効率的な項目選択が可能になるとと思われる。

マルチスケール空間自体に焦点を当てた研究もなされている。Guiardらはマルチスケール空間におけるポインティングがAccotらのSteering Law<sup>19)</sup>でモデル化できることに着目し、マルチスケール空間におけるポインティングのIndex of Difficultyについて説明している<sup>1)</sup>。このようなモデル化によってマルチスケールインタフェースを用いたポインティングのパフォーマンスを定性的に述べる事が可能になると考えられる。

### 3. インタフェースの実装

項目を1次元的に配置することでカーソル操作のみで操作可能なマルチスケールインタフェースを提案する。基本的なコンセプトは、カーソルが項目に近づくとその項目にズームインし項目から離れるとズームアウトするというものである。

最も重要なことは、Fisheye viewのようにディスプレイ上のカーソルの位置とズームインする箇所を1対1に対応付けられないことである。そのような実装を行うと限られたディスプレイ上の領域を配置される項目で分け合うことになり、項目数が大きい場合、一つの項目に割り当てられる領域が極端に小さくなってしまふ。そのような場合、カーソルのわずかな移動に伴う項目位置の変化が発生し目標の項目をポインティングすることは非常に困難である。我々はディスプレイ上のカーソル位置とズームイン箇所を対応付けるのではなく、カーソルの移動に伴ってカーソルの移動方位にある項目に徐々にズームインすることを考えた。そのような実装を行うと、ユーザはスケールが小さくカーソルを正確に目標項目の方向へ移動することは難しい状態ではおおまかに操作を行い、スケールが大きくなるにつれて微調整を行えば良い。

この操作はちょうど、平面上に置かれた対象の位置へ自動車に向かうようなタスクとしても考察すること

ができる。最初は対象の方位へ大まかに走っていき、対象に近づくにつれて微調整を行うことになる。この際、運転中のどの過程でも極端に微妙な操作は発生せず、メンタルには操作の敏感さに変化は生じない。また、最初の向きが少々ずれていても、後に容易に修正することが出来る。この意味で、経路からの逸脱を許容しない場合における法則であるSteering Law<sup>19)20)</sup>よりも制約が緩い。

ズームインの実装上注意しなければならないことは、スケーリングによってディスプレイ上の項目の位置が変化してしまうことである。ユーザが選択目標としている項目がディスプレイ上で移動するとポインティングが困難になる。このような問題はFisheye Viewにおいて報告されている<sup>6)</sup>。ユーザが選択目標としている項目をシステムが正確に把握することは困難だが、ユーザは通常、目標項目に向かってカーソルを移動するため、カーソルの移動方位に配置されている項目の位置がディスプレイ上で移動しないようにスケーリングを行うことでこの問題に対処できると考えられる。

次に項目の配置の仕方について考える。1次元的な項目の配置には、直線状、環状等多数のバリエーションが考えられる。しかし、ズームインする際にカーソルの移動方位の半直線上に複数の項目が存在すると、どの項目にズームインして良いのかを判断することが難しい。そのため、1次元的とはいえ正弦波等の複雑な曲線上に項目を配置するのは望ましくない。本論文では全ての項目に等距離で到達できるように円弧状に項目を配置した。この内、環状になり閉じているものを“Ring”、閉じていないものを“Arc”と呼ぶ。

#### 3.1 “Ring”

“Ring”の実装について述べる。議論を簡単にするために初めに言葉と記号の意味を定義し、その後実装の詳細を述べる。

“Ring”の項目は半径 $r$ の円周上に配置されている。その円の中心を原点とし、スケールとは無関係に固定された空間をインタフェース空間と呼ぶ。インタフェース空間上の項目の位置はスケーリングが起っても変化しない。一方、操作画面上の空間をディスプレイ空間と呼ぶ。ディスプレイ空間上では、項目の位置はスケーリングだけでなく、インタフェース空間上でのディスプレイの平行移動(パン)によっても変化する。なお、インタフェース空間とディスプレイ空間では $x, y$ 両軸の向きは同じであるとする。また、カーソル位置からカーソルの移動方位へ引いた半直線と“Ring”の円周の交点のことをTarget Pointと呼ぶ。カーソルの移動方位にある項目が移動しないようなズームが必

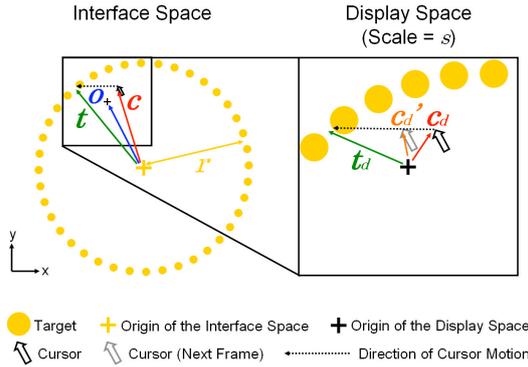


図2 インタフェース空間とディスプレイ空間における位置ベクトル

要だと前述したが、Target Point が項目と項目の間で項目が存在していない可能性も考えられるため、実際には Target Point が連続する 2 フレーム間で移動しないようにスケーリングを行う。なお、カーソルの移動方向は、連続する 2 フレームのディスプレイ上のカーソル位置から求める。

スケールを  $s$ 、全項目が表示されるための最小スケールを  $s_{min}$  とする。なお、カーソルが円周に到達した際のスケールを 1 とし、それ以上スケールは大きくならないものとする。 $s = s_{min}$  のときのディスプレイ空間上での “Ring” の半径を  $r_{min}$  とする。また図 2 に示すように、インタフェース空間上での、ディスプレイ空間の原点、カーソル、Target Point の位置ベクトルをそれぞれ  $o, c, t$  とし、ディスプレイ空間上でのカーソル、Target Point の位置ベクトルをそれぞれ  $c_d, t_d$  とする。また、 $o', c'_d, s'$  等のように ' の付いた値はこれから計算したい次のフレームの値を、 $o, c_d, s$  等は現在のフレームの値を表すものとする。ただし、 $t$  および  $t_d$  は連続した 2 フレームから定義されるので ' 付きの値を定義しない。また、定数である  $r, r_{min}, s_{min}$  についても ' 付きの値を定義しない。

このとき、ディスプレイ空間上のある点の位置が  $x_d$  で、対応するインタフェース空間上での点の位置が  $x$  で表されるとすると、その間には次のような関係が成り立つ。

$$x - o = \frac{x_d}{s} \quad (2)$$

次に、ズームの実装について説明する。ズームアウトはカーソルがディスプレイ空間の原点に近づくにつれて  $s = s_{min}$  に近づけ、また “Ring” のディスプレイ上での中心をディスプレイ空間の原点に近づければ良い。選択操作のユーザビリティに大きく影響するのはズームインの実装であるので、ここではズームイン

について詳しく述べる。

ディスプレイ空間でのカーソルの位置  $c'_d$  はカーソルの移動によって直接的に決定されるが、 $c'$  に関してはスケールの変化とインタフェース空間上でのディスプレイの移動を考慮しなければならない。(2) を考えると  $c'_d$  と  $c'$  の関係は  $s', o'$  を用いて次のように書ける。

$$c' - o' = \frac{c'_d}{s'} \quad (3)$$

次に Target Point について考える。Target Point がディスプレイ上で移動しない、すなわち  $t_d$  が変化しないという制約式は (2) より以下のように表される ( $t$  も変化しないことに注意が必要である)。

$$s'(t - o') = s(t - o) = t_d \quad (4)$$

さらに、カーソルの移動方向はディスプレイ空間上、インタフェース空間上で等しいので以下の式が成り立つ。ただし、 $k$  は Target Point までの到達度を表す変数である。

$$c' = c + k(t - c) \quad (5)$$

(4) を変形すると以下ようになる。

$$o' = t - \frac{s}{s'}(t - o) \quad (6)$$

これと (5) を (3) に代入して整理すると

$$(1 - k)(t - c) = \frac{s(t - o) - c'_d}{s'} \quad (7)$$

となり、さらに (7) は

$$t_d = s(t - o) \quad (8)$$

$$c'_d = c_d + k_d(t_d - c_d) \quad (9)$$

を用いて (ただし、 $k_d$  はディスプレイ上での Target Point までの到達度を表す値である)、以下のように変形できる。

$$(1 - k)(t - c) = \frac{(1 - k_d)(t_d - c_d)}{s'} \quad (10)$$

ここで、

$$t - c = \frac{t_d - c_d}{s} \quad (11)$$

であることを用いると、(10) は

$$\frac{1 - k}{s}(t_d - c_d) - \frac{1 - k_d}{s'}(t_d - c_d) = 0 \quad (12)$$

となる。これを整理すると

$$\left( \frac{1 - k}{s} - \frac{1 - k_d}{s'} \right) (t_d - c_d) = 0 \quad (13)$$

となるので、 $t_d - c_d \neq 0$  であれば以下の関係が得られる。

$$\frac{1 - k}{s} - \frac{1 - k_d}{s'} = 0 \quad (14)$$

ここで未知数は  $s'$  と  $k_d$  であるので、片方を何らかの方法で定めてやればよい。 $s'$  に着目すると、 $s'$  は  $k_d$  に関して単調増加し、 $k_d = 0$  のとき  $s' = s$  であり、 $k_d = 1$  のとき  $s' = 1$  という制約が得られる。これを実現する簡単な方法は  $s'$  を  $s$  と  $k_d$  の関数  $s' = f(s, k_d)$  とすることである。以下、この関数をスケール関数と呼ぶ。最も簡単なスケール関数は  $s'$  が  $k_d$  に対し線形に増加する形で次式のように記述できる。

$$s' = (1 - s)k_d + s \quad (15)$$

しかし  $s'$  が  $k_d$  に対して線形に増加すると、カーソルが中心から円周に向けて動き始めたときに急激な拡大が起こってしまい操作が難しい。そこで、スケールの逆数である  $1/s'$  が線形に減少するように、次式を満たすようにスケール関数を定めた。

$$\frac{1}{s'} = \left(\frac{1}{s} - 1\right)(1 - k_d) + 1 \quad (16)$$

なお、 $t_d - c_d = 0$  が成り立つ場合は、 $s' = 1, k = 1$  とする。

$s'$  が定まれば (14) より  $k$  が、(6) より  $\sigma'$  が求まる。また、(5) に  $k$  を代入して  $c'$  を求めることができる。

### 3.2 “Arc”

“Ring” では、ズームインの前に全体を見るため一度カーソルを中心に戻さなければならない。この操作は項目数が大きい場合にだけ必要であり、当初の目的である項目数の大小によるメンタルモデルの相違を解消できていない。そこで、“Ring” を改良してカーソルを中心に戻さずに操作可能なインタフェース、“Arc” を提案する。

“Arc” のアイデアは、選択操作が開始された時点（例えば、「チャンネル操作」を選んだ後にディスプレイ上に数百のチャンネルが表示され、ユーザがチャンネルの選択操作を開始する時点）でカーソルのある位置を中心とした“Ring”を作成するというものである。この場合ユーザはカーソルを中心に戻す操作を行わなくてよい。しかし、操作開始時にカーソルのある位置を中心とすると、ディスプレイ（またはウィンドウ）から項目がはみ出してしまう可能性がある。そこで“Arc”では、円周上でディスプレイに表示されている部分にのみ項目を配置する。

実装上“Arc”は、“Ring”のディスプレイ空間の原点をカーソルの初期位置に設定し、項目の配置の方法を変更することで実現できる。“Arc”の外観を図3に示す。

## 4. 評価実験

本実験の目的は、提案する“Ring”と“Arc”のユー

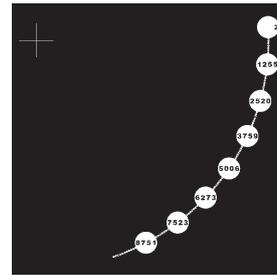


図3 “Arc”の外観

ザビリティ評価を被験者実験を通して行い、その有用性を示すことである。評価は客観的な操作時間と被験者の主観的評価の両面から行った。比較のためにスクロールバーを実装し、スクロールバーによる選択も行った。以下、この比較用スクロールバーをスクロールバー一般と区別して“Scrollbar”と表記する。

### 4.1 実験手順

“Scrollbar”, “Ring”, “Arc”それぞれについて項目数が200, 1000の六つのケースについて実験を行った。

被験者は日常的にスクロールバーを操作しているため、“Ring”や“Arc”と比較して“Scrollbar”は有利であると考えられる。各インタフェースとも十分に習熟した状態で実験を行うために、被験者には実験前に各インタフェースの操作を練習する時間を設けた。このとき、インタフェース操作だけでなくそのインタフェースを用いたポインティングタスク自体（目標項目をできるだけ早く選択する）に慣れてしまう可能性も考えられたので、“Scrollbar”についても同様に練習を行なわせた。練習時間は六つのすべてのケースの合計で20分程度設けた。

実験は以下の手順でおこなった。まず、六つのケースから無作為に一つを選び被験者に与える。被験者は一つのケースについて10回の選択操作を行う。それが終わると次のケースに移る。このとき、一度行ったケースが選択されることはない。六つのケースを一度ずつ行うタスクを1ブロックとし、実験は3ブロック行った。なお、3ブロックを通して同じインタフェース（同じケースではない。項目数200と1000の“Ring”は同じインタフェースであるとみなした）が2回連続して選ばれないようにした。なお、被験者には各ケースの実験を開始する時点で、それがどのケースであるかを示した。すべてのタスクが終了した後に、主観的な評価を行うためアンケートを行った。

### 4.2 タスク

図4にタスクの流れを示す。まず0から9999の中から無作為に選んだ数字を被験者に2.5秒間提示する

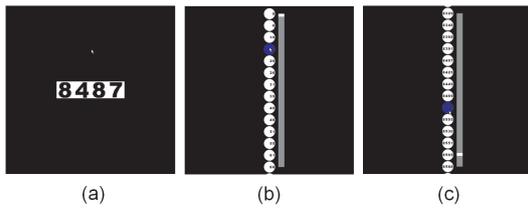


図4 選択タスクの流れ: (a) 被験者が選択する項目の数字を提示する。(b) 数字が消えて選択インタフェースが提示される。(c) 被験者は可能な限り早く選択を行う。

(図4(a)). この間に被験者は数字を記憶する。数字が消えると選択インタフェースが表示される(図4(b)). 被験者は提示されたのと同じ数字が描かれた項目を可能な限り早く選択する(図4(c)). この一連の流れを1回の選択操作とする。(c)が終わると(a)に戻る。

項目上に表示する数字は0から9999から選ばれるが、数字の選び方には意図的に偏りを持たせた。これは、項目の名前や属性(作成日時等)が一様に分布しているとは限らず(例えば辞書に収録された単語のイニシャルは、AからZまで均一に分布しているわけではない)、実際に選択操作を行う際にはその偏りを考慮しなければならない。分布の仕方を被験者が予測できないように、数字の分布は1回の選択操作ごとに変更した。

### 4.3 実験機器

実験は解像度1400x1050の14インチディスプレイを備えたPC3台を用いて並行して行われた。C/D比(マウスの速度)及びMouse Accelerationに関しては、各ユーザが最も使いやすいと感じる状態に設定させた。

### 4.4 被験者

被験者は23歳から26歳までのボランティアの男性6人(平均23.5歳)で全員右利き、右手でマウスを操作した。6人のGUI使用歴、1日あたりのGUIを用いた作業時間はそれぞれ平均して6.2年、6.0時間であった。

### 4.5 実験結果

項目数  $n = 200$  の場合、選択に要した平均時間は“Scrollbar”, “Ring”, “Arc”を用いた場合それぞれ4.27, 4.18, 3.87秒であった。t検定を行った結果, “Scrollbar”-“Arc”間で有意水準0.01, “Ring”-“Arc”間で有意水準0.05で有意差が見られた。“Scrollbar”-“Ring”間では有意差は見られなかった。

項目数  $n = 1000$  の場合、選択に要した平均時間は“Scrollbar”, “Ring”, “Arc”を用いた場合それぞれ6.15, 6.08, 5.90秒であった。有意差はいずれの組み合

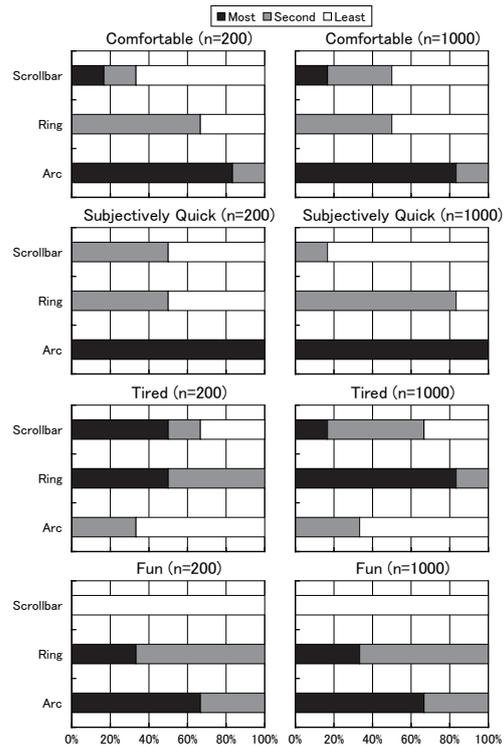


図5 被験者の主観的評価

わせでも見られなかった。

次に主観的評価について述べる。以下のような項目について、 $n = 200, 1000$ のそれぞれの場合について, “Scrollbar”, “Ring”, “Arc”に1位から3位までの順位を付けさせた。

- Comfortable: 項目を快適に選択できた。
- Subjectively Quick: 項目を高速に選択できた。
- Tired: 操作していて疲れた。
- Fun: 操作していて楽しかった。

また、3つのインタフェースを使って感じたことを自由に記述させた。主観的評価の結果を図5に示す。

## 5. 考察

客観的評価では、 $n = 200$ の場合に“Arc”の操作時間が短かったことを除いて有意な差は見られなかった。“Scrollbar”-“Ring”間で有意差が見られなかったことには, “Ring”が一度中心までカーソルを戻すことによって時間をロスしたことが関係していると思われる。 $n = 1000$ においては“Scrollbar”-“Arc”間でも有意差が見られなかった。しかし、主観的には $n = 200, 1000$ 両ケースですべての被験者が“Arc”を用いた場合に最も素早く選択操作を行えたと答えている。このことから項目数の大小によるメンタルモデルの相違を軽減す

るという当初の目的は達成できたのではないかと考えられる。しかし、項目数が増えて有意差がなくなったことから、より多くの項目を扱った場合に“Scrollbar”と比べて“Arc”が有用なのかということには疑問が残る。このようなことが起こった原因として、スケーリング前後のスケール差が大きいとズームインの最終段階で急激にスケールが大きくなり、目標項目をディスプレイ内に留めるのが難しくなることが考えられる。項目数の増加はより正確なポインティングを必要とするため、 $n > 1000$ において“Arc”が有用であるとは言えない。このような問題は、(16)の代わりにズームインの最終段階でよりゆるやかなスケーリングを行う関数を導入することで軽減できるのではないかと考えられる。

快適性に関しては、 $n = 200, 1000$ で共に1人の被験者を除き全員が、“Arc”が最も快適だと回答している。ユーザが快適だと感じることは素早い操作と共に重要であり、この結果からも“Arc”は有用であると考えられる。なお、“Scrollbar”を最も快適と感じた1人からは、普段から使っており慣れているため安心して使えたという意見が得られた。“Ring”や“Arc”もより長時間使うことで評価が良くなる可能性が考えられる。

疲れに関しては、 $n = 200, 1000$ ともに“Ring”が最悪の結果となった。“Ring”は“Arc”と比較してズームインに利用できるカーソルの移動距離が小さく急激なスケーリングが起こりやすい。その結果、“Ring”はユーザに正確なポインティングを要求することになり、“Ring”の操作には集中が必要になる。これが“Ring”の疲れの主な原因ではないかと考えられる。“Scrollbar”と“Arc”を比較すると“Arc”の方が良い結果が得られており、このことから“Arc”は有用であると考えられる。

また、3人の被験者から“Scrollbar”で表示されているのが全体の中のどの辺りなのかわかりにくいという意見が得られた。4.2で述べたように、項目の数字の分布を均一にしなかったため、全体を一度に見渡すことができる“Ring”や“Arc”に比べて“Scrollbar”では数字の分布の様子がわかりにくかったためだと考えられる。また、“Scrollbar”では項目数が大きくなるとサムを移動したときのスクロール量が大きくなりすぎて使いづらいという意見も得られた。

## 6. スケール関数の改善と確認実験

ズームインの最終段階で急激なスケーリングが起こる問題を解決するために、以下のスケール関数を用い

た実装を行った。

$$s' = s^{1-k_d} \quad (17)$$

これは、 $\log s$  (Guiardらはzoom indexと呼んでいる<sup>1)</sup>)が $k_d$ に対して線形に変化するようにした式である。この結果、ズームインの最終段階での急激なスケーリングは軽減された。

時間の関係で適切な評価実験は行えなかったが、筆者の操作で実験したところ、 $n = 1000$ において“Ring”の平均操作時間が5.12秒から4.49秒に短縮された(有意水準0.01で有意差あり)。また、数値的な評価は行っていないが $n = 2000, 10000$ においてもスムーズに目標項目を選択できた。

## 7. 結 論

本論文では、大量のデータの中から目的のものを高速かつ快適に選択するためのインタフェース、“Ring”と“Arc”を提案した。従来インタフェースであるスクロールバーを用いると、項目が少なくスクロールの必要がない場合と比べて項目数が大きくなるとユーザは余分にスクロール操作を要求されることとなり心理的負担が大きい。“Ring”と“Arc”はそのような点に着目し、項目数の大小に関わらず単一のメンタルモデルで操作できるように設計した。ただし、項目数が多い場合に“Ring”を用いると、初めに一度カーソルをディスプレイ中心に戻さなければならないという問題があったため、“Ring”を改良したインタフェース、“Arc”を提案した。被験者実験によってこれらを評価した結果、特に“Arc”に関してはスクロールバーと比べて客観的、主観的に良い評価が得られた。“Ring”に関しても6で述べた実験後の改良によって飛躍的なユーザビリティ向上が見られたため、スクロールバーよりも高いユーザビリティを実現できるのではないかと考えられる。

今後の課題として、様々なスケール関数を用いた場合のユーザビリティに関する詳細な調査や、Guiardらの提案するマルチスケール空間におけるポインティングのIndex of Difficultyを用いた定性的な評価等が挙げられる。また、現在の実装ではズームインに失敗したときに微調整することが難しい。これは、ディスプレイの端にカーソルを近づけると項目が周方向に回転するような機能の実装により解決することが出来る。“Ring”や“Arc”の有用性を示すためには、これらを用いた実アプリケーションの作成も必要であると考えられる。

また、円弧状以外の項目の配置を検討する必要がある。

る。特に項目についての多くの情報（ファイル名，作成日時，種類等）を同時にユーザに提示する場合，余白の使い方が重要になる。例えば，項目を直線状に配置することで効果的に余白を確保できる等，目的に応じて項目の配置を考える必要がある。

### 参 考 文 献

- 1) Guiard, Y., Beaudouin-Lafon, M., Bastin, J., Pasveer, D. and Zhai, S. : "View Size and Pointing Difficulty in Multi-Scale Navigation," AVI 2004, pp.117-124 (2004)
- 2) Fitts, P.M. : "The information capacity of the human motor system in controlling the amplitude of movement," Journal of Experimental Psychology, 47, pp.381-391 (1954)
- 3) MacKenzie, S. : "Fitts' law as a research and design tool in human-computer interaction". Human-Computer Interaction, 7, pp.91-139 (1992)
- 4) Baudisch, P., Cutrell, E., Robbins, D., Czerwinski, M., Tandler, P., Bederson, B. and Zierlinger, A. : "Drag-and-pop and drag-and-pick: Techniques for accessing remote screen content on touch- and penoperated systems," Proceedings of Interact, pp.57-64 (2003)
- 5) Guiard, Y., Blanch, R. and Beaudouin-Lafon, M. : "Object pointing: a complement to bitmap pointing in GUIs," Graphics Interface, pp. 9-16 (2004)
- 6) McGuffin, M. and Balakrishnan, R. : "Acquisition of expanding targets," ACM CHI Conference on Human Factors in Computing Systems, pp.57-64 (2002)
- 7) Kabbash, P. and Buxton, W. : "The "Prince" technique: Fitts' law and selection using area cursors," ACM CHI Conference on Human Factors in Computing Systems, pp.273-279 (1995)
- 8) Grossman, T. and Balakrishnan, R. : "The Bubble Cursor: Enhancing target acquisition by dynamic resizing of the cursor's activation area," ACM Conference on Human Factors in Computing Systems, pp.281-290 (2005)
- 9) Furnas, G.W. : "Generalized fisheye views," Proc. of ACM CHI '86, pp.16-23 (1986)
- 10) Sarkar, M. and Brown, M.H. : "Graphical fisheye views of graphs," Proceedings of ACM CHI ' 92, pp.83-91 (1992)
- 11) Gutwin, C. : "Improving focus targeting in interactive fisheye views," Proceedings of CHI 2002, pp.267-274 (2002)
- 12) Zhai, S., Conversy, S., Beaudouin-Lafon, M., and Guiard, Y. : "Human on-line response to target expansion," ACM CHI Conference on Human Factors in Computing Systems, pp.177-184 (2003)
- 13) Robertson, George. G. and Mackinlay, Jock : "The Document Lens," Proceedings of ACM UIST ' 93, pp.101-108 (1993)
- 14) Perlin, K. and Fox, D. : "Pad: An alternative approach to the computer interface," Proceedings of ACM SIGGRAPH ' 93, pp.57-64 (1993)
- 15) Bederson, B. B. and Hollan, J.D. : "Pad++: A zooming graphical interface for exploring alternate interface physics," Proceedings of ACM UIST ' 94, pp.17-26 (1994)
- 16) Furnas, G.W. and Bederson, B.B. : "Space-scale diagrams: Understanding multiscale interfaces," Proceedings of ACM CHI'95, pp.234-241 (1995)
- 17) Jul, S. and Furnas, G. W. : "Critical zones in desert fog: Aids to multiscale navigation," Proceedings of ACM UIST ' 98, pp. 97-106 (1998)
- 18) Guiard, Y., Bourgeois, F., Mottet, D. and Beaudouin-Lafon, M. : "Beyond the 10-bit barrier : Fitts' law in multi-scale electronic worlds," Proceedings of IHM-HCI 2001, pp. 573-587 (2001)
- 19) Accot, J. and Zhai, S. : "Beyond Fitts' law: Models for trajectory-based HCI tasks," Proceedings of ACM CHI ' 97, pp.295-302 (1997)
- 20) Zhai, S., Woltjer, R. (2003). Human movement performance in relation to path constraint: The law of steering in locomotion. Proc. of IEEE Virtual Reality, pp. 149-156 (2003).
- 21) C. Ahlberg and B. Shneiderman, The Alphalider: A Compact and Rapid Selector," Proceedings of ACM Conference on Computer-Human Interaction, pp. 365-371 (1994)
- 22) 増井 俊之, ブラウジングとキーワード検索を統合した GUI 部品 LensBar, WISS98 講演論文集, pp. 153-158 (1998)
- 23) T.Masui, K.Kashiwagi and G.R.Borden IV, Elastic Graphical Interfaces for Precise Data Manipulation, CHI'95 Conference Companion, pp. 143-144 (1995)
- 24) Y. Ayatsuka, J. Rekimoto and S. Matsuoka, Popup Vernier: a Tool for Sub-pixel-pitch Dragging with Smooth Mode Transition, UIST '98, pp.39-48 (1998)
- 25) GonzaloRamos and RavinBalakrishnan, Zliding: fluid zooming and sliding for high precision parameter manipulation, UIST 2005, pp. 143-152 (2005)
- 26) Caroline Appert and Jean-Danie Fekete, OrthoZoom scroller: 1D multi-scale navigation, CHI 2006 Proceedings, pp. 21-30 (2006)