

重なりあったウィンドウ間を移動可能な マウスカーソル操作手法の提案

山中 祥太^{†1} 宮下 芳明^{†1,†2}

本稿では、マウスカーソルをウィンドウの奥へと潜りこませる操作手法を提案し、そのプロトタイプとなるシステムを実装した。重なり合ったウィンドウ間のファイル移動を行わせる評価実験では、通常のマウス操作よりも早くタスクを完了させることが明らかとなった。

A Mouse Cursor Operation for Overlapped Windowing

SHOTA YAMANAKA^{†1} and HOMEI MIYASHITA^{†1,†2}

In this paper we propose an operation method for overlapped windowing; a method that the user slides the mouse cursor under the windows in a simple way. We developed the system and conducted an evaluation experiment, and it revealed that the user can transfer the files between the windows faster than conventional style of GUI.

1. はじめに

例えば図1のように、手前では動画プレイヤーが起動し、奥では動画ファイルの並んだフォルダが開かれている状態があるとき、プレイヤーに隠れている位置の動画ファイルを開いて閲覧するためには次のような操作手順が考えられる。

^{†1} 明治大学 理工学研究科 新領域創造専攻 デジタルコンテンツ系
Program in Digital Contents Studies, Program in Frontier Science and Innovation, Graduate School of Science and Technology, Meiji University
^{†2} 独立行政法人科学技術振興機構, CREST
JST, CREST

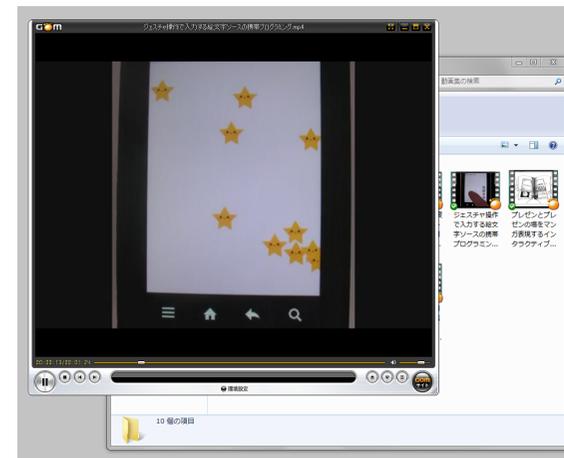


図1 手前に動画プレイヤー、奥にフォルダが開かれている状態

- (1) 奥のウィンドウのクリックして最前面に出す
- (2) 目的の動画ファイルをプレイヤーにドロップする
- (3) プレイヤーをクリックして最前面に出す

この場合、(1)の操作をした時点でプレイヤーの一部がフォルダに隠れてしまうため、(3)のようにして最前面のウィンドウを切り替えるか、あるいはウィンドウ同士が重ならないように位置を調整する必要があり、マウス操作、画面の状態共に煩雑なものとなる。Alt+Tabキーでその都度適切に最前面のウィンドウを切り替えてもよいが、作業で主にマウスを使用する場合には、マウスのみで目的の操作をスムーズに行える方が好ましい。そこで本稿では、従来の操作方法による煩雑さの原因について述べるとともに、このような状況に対応できる新しいマウス操作手法を提案する。さらに、本提案手法におけるシステムを実装し、それをを用いた実験と考察を行う。

2. 従来のウィンドウ表示とマウスカーソル操作の問題点

複数のウィンドウをディスプレイ内に表示する方法は種々あるが、現在広く採用されているのはオーバーラップウィンドウ方式である。これはウィンドウが3次元のレイヤ構造を持つものであるが、マウスカーソルはこのレイヤ構造とは関係なく常に最前面に表示され、2次元の移動しか行えない。それ故に、ウィンドウによって隠れた部分は操作できず、不便だ

と感ずることがある。その他のウィンドウ表示方法として、ディスプレイ内にウィンドウを敷き詰めるタイルウィンドウ方式があるが、個々のウィンドウサイズが小さくなることから視認性に問題が生じ、オーバーラップウィンドウ方式に比べるとあまり利用されていない。

また、ウィンドウフォーカスの問題もある。ウィンドウフォーカスとは、キーボードやマウスからの入力を受け付けるウィンドウを示すものであり、フォーカスの当たっている状態を「ウィンドウがアクティブである」という。Windows7 OS では、クリックしたウィンドウにフォーカスが当たる方式 (Focus Follows Click, FFC) が採用されているが、この方式ではアクティブになったウィンドウが最前面に出るため、そのウィンドウに重なっていた手前のウィンドウが隠れる問題が生じる。現在の作業と全く別のことをする場合にはそのように最前面に出して然るべきであるが、前述のように奥のファイルに対して僅かな操作をする場合であっても、その度にウィンドウの前後関係を変更しなければならない。FFC の他に、マウスカーソルの座標によってフォーカスが移る方式 (focus follows mouse, FFM) があるが、マウスを不意に移動させると作業が妨げられるという問題があり、Windows や Macintosh で採用されているのは前者のクリック追従方式 (FFC) である。

このように、現在一般的に用いられているウィンドウ表示方法とマウスカーソル操作方法はいずれも問題点を抱えている。そもそも、奥にあるものに対して操作を行いたいときに本来は必要のないウィンドウ操作が求められることは、オーバーラップウィンドウ方式のユーザインタフェースにおける重大な問題である。

3. 提案手法

本稿で提案するのは、目的のウィンドウまでマウスカーソルを奥へと移動させて操作する手法である。オーバーラップウィンドウ方式においてウィンドウがレイヤ構造をなしていることは前章で述べたが、マウスカーソルもこのレイヤ構造に組み込むというものである。従来のマウスカーソルはウィンドウをクリックして最前面化するという機能を持っていたが、本提案手法はマウスカーソルがディスプレイ内を 3 次元移動し、マウスカーソル自身が奥のファイルやウィンドウに触れに行くような直感的操作を実現する。

なお、この操作は通常のウィンドウ切り替え方法と共存し、必要なときのみ利用可能なものになっている。よって、完全に別の作業を行う際にはウィンドウ切り替えをし、奥のウィンドウで僅かな操作を行う際には本提案手法を利用するというように、場面に応じた柔軟な操作が可能になる。また、操作を行うために専用のデバイスを必要とせず、あくまで一般的なマウス (左右ボタンとホイールを持ったもの) を用いた操作でカーソルを奥方向へ移動さ

せる。

4. システム

本システムは、ウィンドウ処理部とマウスカーソル処理部に分けて構成されており、この 2 つが連動して動作することによって本提案手法を実現している。また、視覚的・聴覚的な効果によって擬似的な触覚を提示する処理を行っている。以下では、システムの説明で使う用語を定義した上で、具体的な実装方法について述べる。

4.1 本稿で用いる用語の定義

4.1.1 レイヤ

デスクトップを最背面としてウィンドウが重なり合っている状態のとき、最前面にあるウィンドウは 0 番目のレイヤにあり、その n 個後ろのウィンドウは n 番目のレイヤにあるものとする。また、 n 番目のレイヤにあるウィンドウを「 n 番目のウィンドウ」と呼ぶこととする。

4.1.2 同時押し

本システムでは、マウスの左右のボタンのイベントは常時フックされている。左右ボタンが両方とも押下された場合に、先に押下された時刻ともう一方が押下された時刻の差を求め、一定時間 (2ms) 以内であったか判定する。判定が真の場合には、左右のボタンが同時に押下されたものと考え、これを同時押しと呼ぶこととする。この後、左右どちらかのボタンが離された時点で同時押しが解除される。判定が偽の場合は、同時押しではないとして、フックしていたボタン押下のイベントを送出する。

4.1.3 衝突

ウィンドウとマウスカーソル、またはウィンドウ同士の座標を用いて当たり判定を行う。この判定結果が真のとき、それらは衝突していると呼ぶこととする。

4.1.4 対象ウィンドウ

マウスカーソルが乗っているウィンドウを対象ウィンドウと呼ぶこととする。ただし、マウスカーソルよりも手前のレイヤのウィンドウは含めずに考える。すなわち、マウスカーソルが n 番目のレイヤにいるとき、 n 番目以降のウィンドウでマウスカーソルと衝突している最も手前のウィンドウを指す。

4.2 マウスカーソル処理部とウィンドウ処理部

本システムは、起動している各ウィンドウのタイトル、ハンドラ、座標、サイズ、スタイル (透過度や最前面表示などの設定) を取得して、前面のものから順に配列に格納し、以後

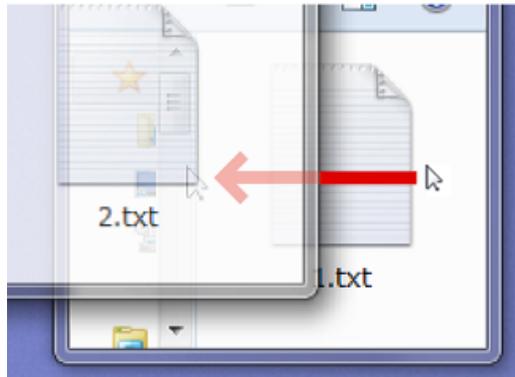


図2 マウスカーソルが潜りこむ様子

これらを監視し続ける。また、その配列における対象ウィンドウのインデックスを保持する。

同時押しをすると、マウスカーソルは対象ウィンドウと同一のレイヤにいるものとして扱われる。よって図2のように0番目と1番目のウィンドウがあるとき、1番目のウィンドウ上で同時押しをすると、マウスカーソルは1番目のレイヤに移動する。つまり、マウスボタンの左右同時押しはカーソルの奥方向への移動へ充てられている。同時押しをしたままマウスを左へと動かすと、カーソルは0番目のウィンドウの裏側へと潜り込んだように表示される。さらにウィンドウが重なっている場合には、同時押しをしたままウィンドウの裏側へマウスカーソルを移動させることで、順に深いレイヤへと移動していく。

同時押しした時点で、対象ウィンドウよりも手前にあるウィンドウを最前面化設定する（設定が解除されるまで常に最前面にある状態にする）とともに、マウスイベントを透過するように設定される。これにより、今後クリックやドラッグをしても対象ウィンドウより手前のウィンドウには操作が行われず、かつ対象ウィンドウにマウスイベントが起こってもウィンドウの前後関係は変更されない。

同時押しをしていないときには、マウスカーソルは常に手前方向に移動しようとする。つまり、図2において、0番目のウィンドウの裏側で同時押しを解除し、マウスカーソルと0番目のウィンドウが衝突しない位置まで動かせば、マウスカーソルは最前面に戻ってくる。マウスカーソルの前面にウィンドウがある状態で同時押しを解除しても、マウスカーソルは手前方向に移動しない。よってマウスカーソルはそのウィンドウの1つ奥のレイヤに存在し続ける。すなわち、図2でマウスカーソルが左の位置にあるとき、マウス操作は1番目の

ウィンドウに対して行われる。以上をまとめると、マウスカーソルの前後移動については以下のように動作することになる。

- 通常時：マウスカーソルは常に手前方向へ移動する。マウスカーソルが最前面に来るか、現在マウスカーソルがいるレイヤよりも手前のウィンドウに当たると、そのレイヤにとどまる。
- 同時押し時：マウスカーソルは奥方向へ移動する。マウスカーソルは対象ウィンドウと同一のレイヤにいるものとして扱われる。

ウィンドウの透過度は、現在は半透過（20～25%程度）を採用している。完全透過させる方法も考えられるが、平岡の研究¹⁾によると、アクティブウィンドウ以外をタイトルバーのみの表示にするとそれらを見失うと指摘されており、ウィンドウの概形を把握可能な半透過にとどめることにした。

4.3 視覚・聴覚的触覚提示部

3次元デスクトップ環境でアイコンやウィンドウにマウスカーソルが触れると触覚を提示するシステム²⁾では、触覚提示があることによって奥行きが分かりやすいとの評価がされている。また、ばらばらウィンドウ³⁾の評価実験において、ウィンドウの消去・表示に合わせてクリック感がほしいという意見があったと述べられている。これは本提案手法においてマウスカーソルがレイヤを移動するタイミングに相当する。よって本提案手法においても、ユーザが奥行きを理解しやすくするために触覚提示が必要と考えられる。しかし、本研究では3章にあるように、あくまで一般的なマウスのみを用いた操作を提案したい。そこで、マウスカーソルがレイヤを移動したことをユーザに提示するために、視覚効果と音響効果による擬似的な触覚提示方法を利用する。視覚効果による触覚の描出効果は渡邊らによって確かめられており、音による影響も示唆されている⁴⁾。本システムでは、マウスカーソルがレイヤを移動すると「マウスカーソルとウィンドウがぶつかる」と捉えることとする。このとき、金属のぶつかる音を鳴らし、マウスカーソルの画像を一瞬だけ縮小・拡大することで段差を移動したような触覚を提示しようと考えた。また、触覚提示とは異なるが、同時押しまたは左ボタンを押下しているときには、図3のようにマウスカーソルを小さく表示し、マウスカーソルを押し付けている感覚が理解しやすいようにした。

5. 操作の例

本章では、本提案手法が有効に機能する例を挙げる。なお、操作環境はWindows7を想定している。



図 3 通常時のマウスカursor (左) と、押しつけた状態 (右) のサイズ比較



図 4 複数のウィンドウの奥に潜ってデスクトップに到達する様子

5.1 デスクトップのアイコンをダブルクリックする場合

図 4 のように、ウィンドウが複数ある状態でマウス操作をする作業中に、デスクトップ上のファイルやアプリケーションのショートカットなどをダブルクリックして起動する状態を考える。通常であれば、手前のウィンドウを移動したり全て最小化させたりしてから目的のアイコンをダブルクリックするが、操作量が多くなってしまふ。これに対し本提案手法では、デスクトップ上で同時押しをして目的のアイコンをダブルクリックするという一連の操作のみで完了する。

5.2 奥のウィンドウを操作すると手前のウィンドウが完全に隠れる場合

図 5 のように、手前のブラウザよりも奥のフォルダのウィンドウサイズが大きい場合 (奥のウィンドウが最大化されている場合も含む) に、奥のフォルダ内の PDF ファイルをダブルクリックする場面を考える。通常は奥のフォルダを 1 度クリックした時点で最前面化して

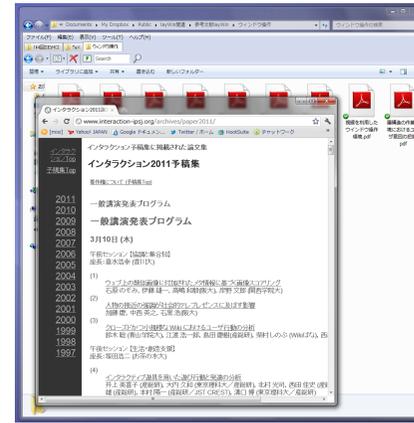


図 5 奥のウィンドウを操作すると手前のウィンドウが完全に隠れてしまう配置

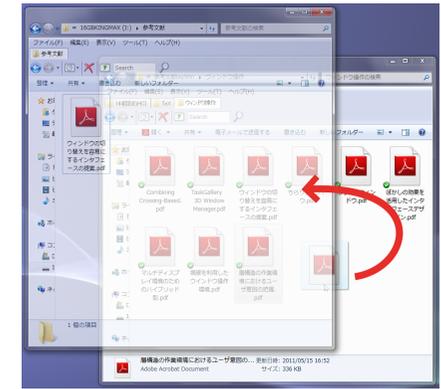


図 6 奥の隠れたファイルを手前のフォルダにドラッグアンドドロップする様子

しまい、ブラウザが完全に覆い隠されてしまうため、タスクバーにあるブラウザのアイコンをクリックして再度最前面化するか、フォルダを移動させるといった操作が必要になる。本提案手法では、奥のウィンドウを最前面化せずに PDF ファイルを開けるため、ブラウザを最前面に戻す操作が必要なくなる。

5.3 手前と奥にあるフォルダ間でのドラッグアンドドロップ操作

図 6 のように 2 つのフォルダを開いており、奥の隠れた位置にあるファイルを手前のウィンドウへドラッグアンドドロップする場面を考える。通常であれば、奥のウィンドウをクリックして最前面化したり、手前のウィンドウを移動したりして、目的のファイルと移動先のフォルダが同時に見えている状態にしてからドラッグアンドドロップする必要がある。本提案手法では、ウィンドウに対する操作をせずに直接目的のファイルやフォルダのある位置へマウスカursorを移動させることができるため、ドラッグアンドドロップ操作が一連の動きで可能である。

6. 実験

本提案手法の有効性を確かめるために、被験者が簡単なファイル移動のタスクを完了するまでの時間を計測する実験を行った。以下で実験の詳細について述べる。

6.1 実験方法

異なるフォルダ間でのファイル移動を行うタスクを被験者に与え、操作を完了するまでの時間を計測した。被験者は大学生及び大学院生の合計6名で、普段PCの操作に用いるのは、マウス、トラックボール、タッチパッドなど、被験者によって様々である。今回の実験では、操作にマウスのみを用いることとし、既存の手法と本提案手法（マウスカーソルがウィンドウの下に潜る）とでタスクを5回ずつ行わせた。実験を行うシステムの順序は被験者毎に変更した。実験で認めた操作は、ファイルの移動、ウィンドウの移動、ウィンドウの最大化・最小化、ウィンドウのサイズ変更、タスクバーに表示されたウィンドウをクリックすることによる最小化と前面移動である。被験者には可能な限り早くタスクを行うように伝えた。

実験開始時のフォルダの配置を図7のスクリーンショットに示す。また、移動前と移動後のファイル配置のイメージを図8に示す。デスクトップ上に3つのフォルダA, B, Cが左から並んでおり、テキストファイル1, 2, 3がフォルダB, C及びデスクトップに置かれている。マウスカーソルはフォルダAの中央に配置される。この状態から、3つのテキストファイルをのように指定したフォルダへ移動するまでを1回のタスクとした。なお、ここでは便宜上デスクトップも1つのフォルダと見なす。

被験者には、本提案手法によるマウスカーソル操作に慣れさせるため、潜る機能ありでフォルダ間のファイル移動を10~15分程度行わせた。また、本システムの仕様を説明しながら、潜る機能を使って素早く操作するためのアドバイスを適宜行った。その後、実験と同じウィンドウの配置でファイルを移動する練習を、潜る機能なし・ありの両方で行わせた。これは、実験を開始してからファイルを移動する順序やウィンドウを移動する位置などを試行錯誤しないようにし、操作をスムーズに行った場合の時間を比較するためである。さらに、実験時にファイルを誤ったフォルダに移動してしまったり、意図しないウィンドウの最大化を行ったりした場合にも、同様の理由でタスクのやり直しを認めた。なお、ウィンドウの透過率は筆者らが実際にタスクを行って操作しやすいと感じた23%に固定して行っている（実際にはユーザによって調整可能である）。

6.2 結果と考察

被験者毎の操作時間を表1に示す。また、被験者毎に平均操作時間を比較したグラフを図9に示す。全被験者について、本提案手法の方が操作時間が短縮される結果となった。平均操作時間を被験者毎の対応あり両側t検定にかけたところ、有意水準1%で有意に操作時間が短縮されたことが確認できた。

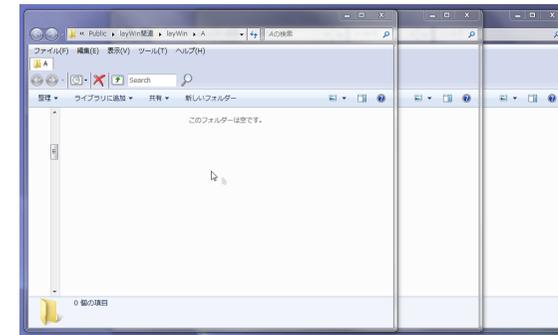


図7 実験開始時のウィンドウの配置

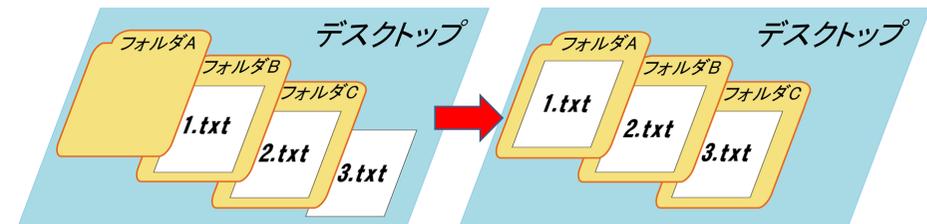


図8 実験開始時のフォルダとファイルの配置（左）と移動後の配置（右）

表1 潜りなし・ありでの被験者毎の操作時間（秒）

回数	被験者1		被験者2		被験者3		被験者4		被験者5		被験者6	
	なし	あり										
1回目	10.0	7.8	7.3	5.2	7.3	7.2	7.0	5.9	7.5	6.2	7.2	6.9
2回目	7.7	7.2	7.0	4.5	8.5	7.4	7.3	6.0	8.1	6.0	7.5	6.1
3回目	8.5	6.6	6.3	4.4	9.5	6.6	7.3	6.1	8.4	5.1	7.2	6.9
4回目	10.4	7.2	6.8	5.9	8.5	6.2	8.2	6.9	7.3	5.1	8.8	6.2
5回目	9.0	5.7	6.4	5.1	7.9	5.8	8.2	5.9	7.1	6.3	7.2	6.0
平均	9.1	6.9	6.8	5.0	8.3	6.7	7.6	6.1	7.7	5.7	7.6	6.4

実験後に5段階評価のアンケートを実施し、本システムのユーザビリティについて尋ねた。質問項目と回答結果を表2に示す。さらに、「今回はどちらがタスクをこなしやすいかったか」という質問に対しては6名全員が潜る機能ありと回答し、「今後、潜る機能ありのマウスカーソルを使いたいと思うか」という質問に対しても全員が「そう思う」と回答したこ

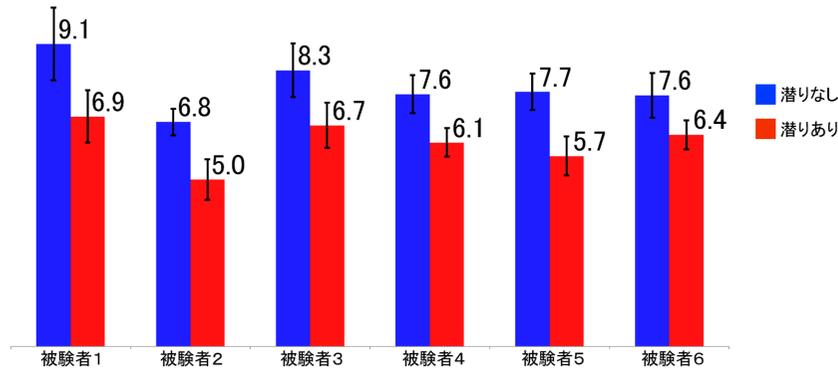


図9 被験者毎の平均操作時間(秒)の比較

表2 アンケート項目と回答の平均値

質問項目	回答
左右ボタン同時押しによって潜る、という操作のしやすさ	3.0
常に手前のレイヤに移動しようとするシステムの操作しやすさ	3.5
潜った先での(ウィンドウの裏側での)マウスカーソル操作のしやすさ	3.5
目標のウィンドウへの辿り着きやすさ(意図したレイヤでマウスカーソルを止められるか)	3.8
音によるレイヤ移動の分かりやすさ	4.5
マウスカーソルの拡大・縮小によるレイヤ移動の分かりやすさ	2.8
潜って操作するという機能の便利さをどのくらい感じたか	4.5
潜る機能ありの場合の、操作したいファイルへの辿り着きやすさ	4.5
潜る機能なしの場合の、操作したいファイルへの辿り着きやすさ	2.0

とから、本提案手法が好意的に受け止められていると考えられる。

実験を通しての所感を自由記述で求めたところ、潜っている最中の階層を把握しづらい、手前のウィンドウの文字が重なって対象ウィンドウが見づらいといった意見が得られた。マウスカーソルがレイヤを移動する際のマウスカーソルの拡大・縮小は見えなかったという回答もあり、狙いであった視覚効果による触覚提示をうまく扱えなかった。この点については、マウスカーソルの拡大・縮小率を上げたり、やや時間をかけて見せたりすることで解決を図りたい。一方、金属音によるレイヤ移動については好評であり、聴覚提示を用いた移動の提示は本提案に有効に作用することが示唆された。そのため、今後のシステムにも積極的に活用していく予定である。今回は左右ボタン同時押しで潜る操作を行ったが、普段行わない動作なので非常に違和感を覚えたという意見があった。それとは逆に、操作の覚えづらさ・

操作の行いづらさはなかったという感想もあった。左右のボタンを押下するタイミングがずれるとマウスドラッグになってしまう問題を指摘されたため、同時押しの判定を2ms以内とするのではなく、ユーザが任意に値を設定できるようにすることを検討している。

7. 改善すべき点

本提案手法に関して得られた意見のうち、改善すべき項目と筆者らの考えを述べる。

7.1 マウスカーソルを任意のレイヤで止められない

「システム概要」で述べたように、マウスカーソルはウィンドウに衝突したときにそのレイヤに移動する、というアルゴリズムで動作している。冒頭の図1のようにウィンドウが配置されている場合には本提案手法が有効に働くが、奥よりも手前のウィンドウのサイズが大きい場合には操作が行いづらい。マウスカーソルを一旦デスクトップまで押し付けて潜りこんでから目的のウィンドウまで手前へ移動させていくことも可能だが、操作が煩雑である。また、ウィンドウを最大化して作業している場合も同様であり、こちらはそもそもマウスカーソルが奥へ移動するための隙間がない。そこで、ディスプレイの端で同時押しをしている場合にはデスクトップのレイヤまでマウスカーソルを移動させるか、あるいは同時押ししている最中は全てのウィンドウサイズを若干小さくしたり、マウスジェスチャでレイヤ移動を行ったりする仕様を検討している。

7.2 潜ったレイヤでの操作が行いづらい

2, 3個程度のウィンドウの下に潜った場合は容易でも、それよりも多くなると操作しづらいとの意見が多くあった。具体的には、重なったウィンドウが同系色だと操作対象のウィンドウが把握できない、ウィンドウが白いとマウスカーソルがどこにあるのか見えない、などという問題である。これに対しては、対象ウィンドウを赤く縁取りすることで認識しやすくしたり、マウスカーソルの周囲だけウィンドウに穴を開けたように表示することで解決するという方法や、手前のウィンドウを半透過ではなく枠だけ表示する、手前の2, 3枚は半透過でそれ以降は完全透過にする、奥のウィンドウほど透過度を上げる、といった方法を検討している。

7.3 3次元移動が可能になることで認知的負荷が増加する

本提案手法による認知的負荷は非常に根本的な問題であり、マウスカーソルを奥方向や手前方向へ移動させるべきかどうかという議論になる。可能な限り、ユーザに負荷のかからない操作方法にしていきたい。例えば、マウスカーソルがどの程度奥へ移動したのかを示す方法として、棒グラフや数値を表示する手法も考案したものの、認知的負荷がかかる懸念が

あったため、今回実装したシステムでは擬似触覚を利用してレイヤ移動を提示する程度に留めている。

7.4 レイヤを移動するために、ウィンドウを迂回する必要がある

本提案手法による操作では、ウィンドウを回りこむようにカーソルを動かす必要があるため、マウスの移動量が多くなるという問題を抱えている。そこで、この問題に対してもマウスジェスチャでのレイヤ移動をするか、マウスの右ボタンを押しながらホイール回転を行うといった対処をしようと考えている。

また、今回のシステムの仕様ではユーザの意図しないタイミングでマウスカーソルが手前に移動してしまうことがある。具体例を挙げると、潜った先の対象ウィンドウに表示されている文章を読むためにマウスカーソルを移動させると、手前のレイヤに移動してしまう、といったことである。そこで、手前に戻るための操作を用意するのはどうかという提案をされた。これによって誤操作は減少すると思われるが、その分だけ操作が煩雑になるので、慎重に検討したい。

7.5 奥のウィンドウ内の、隠れていない部分を操作する場合

冒頭の図1において、動画プレイヤーに隠れているファイルではなく、既に見えているファイルをダブルクリックする場合には、本提案手法は有効に働かず、どうしても一度動画プレイヤーが背面に移動してしまうことになる。このように、本提案手法の有効性はウィンドウの重なり方やサイズに大きく依存してしまう。よって、どのようなウィンドウの配置やサイズであっても対応可能なシステムにしたいと考えている。

8. 今後の課題と展望

今回実装したシステムを元に、マウスジェスチャに対応したシステム、潜ったウィンドウの表示を変更したシステムなどの検討すべきバージョンを実装し、評価を行っていく。また、奥のレイヤへ潜る際にマウスカーソルの先端をウィンドウに差し込むようなグラフィックにするかどうか、マウスジェスチャ対応バージョンではその場でウィンドウを突き破ることで奥や手前に移動するといった視覚的効果を加え、レイヤを移動したことをより直感的に理解できるようにする。また、本システムはポインティングデバイスとしてマウスを対象にしたものだが、本提案手法はタッチパッドやトラックボールなどでも有効であると考えており、それぞれ適した操作方法を追求していきたい。

今後はタブレット型PCやスマートフォンなどの小さなディスプレイ内で起動を行うことが多くなると考えられる。これらのデバイスで多数のアプリケーションを起動する場合、複

雑さや煩雑さのない動作で操作対象を切り替えられる手法が求められるであろう。そこで、タッチ操作をするデバイスにおいて、押下した圧力を感知することで現在表示されていないアプリケーションを操作するという手法を考案している。このように、最前面に表示されているものよりも奥にあるものを操作可能にするという手法が、プラットフォームに関わらず有効であることを示していきたい。

9. 関連研究

ウィンドウが重なることで発生する問題については様々な解決方法が試みられている。ウィンドウ切り替えの煩雑さを解決するために、加藤らはスライダを用いてウィンドウを手前から順に非表示にするばらばらウィンドウを提案した³⁾。神原らのちらりウィンドウは、ジョイスティックを操作することで視点を移動させ、前面のウィンドウに隠された部分も閲覧可能にするシステムである⁵⁾。久納らは、カメラを用いて実世界での視点移動をディスプレイ内に反映させるシステムを提案した⁶⁾。大野は、視線を利用してウィンドウの移動や最大化、最小化などを行う手法を提案した⁷⁾。これらはいずれもウィンドウ操作のデバイスを必要としているものの、本研究と非常に近い問題設定をしている。また、柴田らは複数のウィンドウを一括することによってマルチタスクを支援する手法を提案しており、タイルウィンドウ方式による効率的なウィンドウ操作を実現している⁸⁾。

RobertsonらのTask Gallery⁹⁾は、仮想的な3次元空間表示を利用することでウィンドウの重なりをなくし、更にウィンドウの配置が記憶されやすくしたシステムである。大内らは、3次元デスクトップ環境においてマウスが2次元でしか操作できないことを問題視し、3次元入力可能なデバイスを使用する手法を提案した²⁾。このシステムでは奥行きを表現するためにマウスカーソルが奥へ移動するほど小さく表示したり、ウィンドウやアイコンなどにマウスカーソルが接触すると触覚を提示したりするといった、本提案手法に似た機能を持っている。

小林らは、オブジェクトをドラッグしている最中にマウスホイールを下回転させることで、マウスカーソル直下のウィンドウを手前から順に非表示にする手法を提案している¹⁰⁾。このシステムは、通常のスクロール操作との競合を避けるために、利用可能な場面をドラッグ操作中に限定している。

Dragicevicは、マウスジェスチャでウィンドウを紙のようにめくることによって、重なったウィンドウ間のドラッグアンドドロップを可能にするシステムfold-and-dropを提案している¹¹⁾。本研究での提案手法がマウスカーソルの位置するレイヤを移動するのに対し、

fold-and-drop はウィンドウ側のレイヤを変更するという手法をとっている。このシステムも、ウィンドウの前後関係を変更せずに別のレイヤにあるウィンドウへの操作を行うことが可能である。

中村らは2つのマウスを用いてウィンドウを操作する環境を提案している¹²⁾。また五十嵐らは、マウスを両手に持って操作することにより、画面内のビットマップ画像を掴んで伸ばしたり曲げたりするインタラクションが可能にするシステムを提案している¹³⁾。

10. ま と め

本稿では、オーバーラップウィンドウ方式においてマウスカーソルをウィンドウの奥へと潜りこませる操作手法を提案し、そのプロトタイプとなるシステムを実装した。このシステムを用いた実験では、通常のマウス操作よりもタスクを早く完了させられることが分かり、本提案手法の有効性を示した。また、この手法が有効に働く事例と改善点を取り上げた。問題点も既に多く判明しているため、今後はこれらを解決する方法についても十分に検討していく。また、操作方法に関してはいくつもの案があり、それぞれの利点・欠点があるため、それらを考慮した上で実装し、評価実験を通してシステムを改善していきたい。

参 考 文 献

- 1) 平岡 茂夫：ぼかしの効果を活用したインターフェースデザイン，福岡工業大学情報科学研究所所報 Vol.12，pp.7-13(2001).
- 2) 大内 勇佑，西野 浩明，宇津宮 孝一，賀川 経夫：触感提示機能を有する3次元デスクトップ環境の開発，火の国情報シンポジウム 2011(2011).
- 3) 加藤 直樹，小國 健：ばらばらウィンドウ：ウィンドウの切り替えを容易にするインタフェース，インタラクション 2003 論文集，pp.123-130(2003).
- 4) 渡邊 恵太，安村 通晃：RUI: Realizable User Interface カーソルを用いた情報リアライゼーション，第27回ヒューマンインタフェース学会研究会「VRの心理と生理」，ヒューマンインタフェース学会研究報告集，pp.35-38(2004).
- 5) 神原 啓介，安村 通晃：ちらりウィンドウ：隠れたウィンドウを覗き見る，インタラクション 2004 論文集，pp.47-48(2004).
- 6) 久納 章寛，岡本 壮平，武藤 直美，中島 誠，伊藤 哲郎：層構造の作業環境におけるユーザ意図の把握，FIT2002 情報科学技術フォーラム 情報技術レターズ，pp.199-200(2002).
- 7) 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).
- 8) 柴田 博仁，大村 賢悟：ウィンドウをドッキングすることによるマルチタスキング

支援，インタラクション 2011 論文集，pp.391-394(2011).

- 9) George Robertson et al.: The Task Gallery: A 3D window manager, In Proceedings of CHI '00, pp.494-501(2000).
- 10) 小林 正朋，五十嵐 健夫：活用：マウスホイール，インタラクション 2005 論文集，pp.175-176(2005).
- 11) Pierre Dragicevic: Combining Crossing-Based and Paper-Based Interaction Paradigms for Dragging and Dropping Between Overlapping Windows. In Proceedings of the 17th annual ACM Symposium on UIST'04, pp.193-196(2004).
- 12) 中村 聡史，塚本 昌彦，西尾 章治郎：活用：2つのマウスを用いたウィンドウ操作機構の設計と実装，情報処理学会研究報告，Vol.99，No.35，pp.1-6(1999).
- 13) 五十嵐 健夫，Tomer Moscovich，John F. Hughes：両手マウスを利用した2次元形状の操作，インタラクション 2006 論文集，pp.57-58(2006).