

PCを用いた作業の効率化を目指したウィンドウ透明化手法 A Window Clearing Method for Efficient Work with PC.

谷川 美沙紀[†] 村田 和義[†] 渋谷 雄[†]
Misaki Tanigawa Kazuyoshi Murata Yu Shibuya

1. はじめに

PCのディスプレイ上に複数のウィンドウを開いているときに、奥のウィンドウやデスクトップを一時的に参照・操作したいときがある。たとえば、統計分析結果を見ながら文書作成をしているときに、届いたメールを見てスケジュールを更新する場合などである。しかし、複数のウィンドウを用いて作業している場合、一時的に他のウィンドウを前面に表示すると、元の作業に戻るために、再び複数のウィンドウを前面に移動し直す必要があり手間がかかる。そこで本研究では、奥にあるウィンドウの参照・操作ならびに元の作業状態への復帰を容易にすることを目的とし、一時的に前面にあるウィンドウを透明にすることで、奥のウィンドウを参照・操作する手法を提案する。

PCを用いた作業中には、ウィンドウの切り替えは頻繁に行われ、1日に数百回も行われる[1]。そのため、ウィンドウの切り替えを効率的に行うことで、PCを用いた作業を効率的に行うことができると考えられる。ウィンドウの効率的な切り替え手法として、Push-and-Pull Switching[2]、ばらばらウィンドウ[3]などがある。Push-and-Pull Switchingは、重なりのないウィンドウを自動的に検出してグループ化し、ウィンドウを切り替える際にグループを切り替える手法である。しかし、グループ化は自動的に行われるため、必ずしもユーザの意図通りにならない。また、ばらばらウィンドウは、ペダルやレバーを用いて、手前にあるウィンドウを順に透明にすることで、奥にあるウィンドウを参照する手法である。しかし、ペダルやレバーといった特殊なデバイスが必要であるという問題がある。

2. 提案手法

提案手法ではマウスのみを用いて、図1のように、前面にあるウィンドウを奥行き方向の重なり順に1枚ずつ透明にしたり透明化を解除したりすることができる。これにより、ウィンドウの配置や上下関係を維持したまま一時的に奥のウィンドウを参照・操作することができる。すなわち、奥にあるウィンドウをメインの作業ウィンドウの上に重ねて表示することなしに奥のウィンドウでの作業を行うことができる。また、元の作業ウィンドウの配置や前後関係を保持しているため、透明化を解除することで元の作業に簡単に復帰できる。

また、提案手法では、一度にすべてのウィンドウの透明化を解除できる機能も用意する。これにより、元のウィンドウ状態に一度の操作で復帰することができる。

さらに、提案手法では任意のウィンドウの奥行き方向の位置を変更できるようにする。これにより、一時的に不要なウィンドウを奥に移動させることや、他のウィンドウの配置や前後関係を崩さずに、特定のウィンドウを

別のウィンドウに近づけて見比べることができる。

なお、提案手法では、探しているウィンドウを明確に覚えていない場合であっても、手前のウィンドウを透明化していくことにより、各ウィンドウが実際の位置に実サイズで表示されるため、目的のウィンドウの再認が容易になるという利点がある。

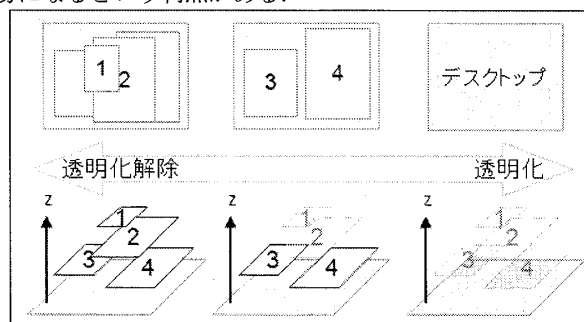


図1: ウィンドウ透明化手法

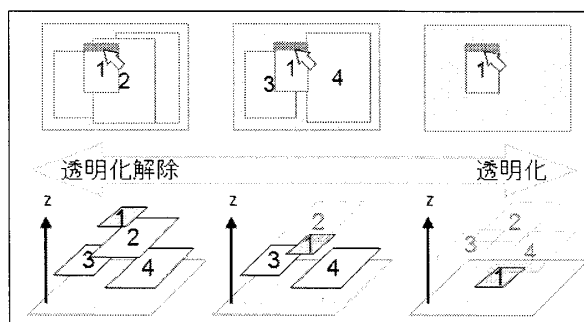


図2: 任意のウィンドウの移動

3. 実装

提案手法は Windows7 上で C++を用いて実装した。なお、実装にあたっては、特殊な装置を必要としないこと、現状の GUI における操作との共存が可能であることを条件とした。そこで、提案手法の操作は、マウスの右ボタンとミドルボタンおよびホイールを用いて行うことができるようにした。右ボタンを押したままホイールを奥に回転することで、前面にあるウィンドウを1枚ずつ透明化することができる。透明化されたウィンドウがある場合、右ボタンを押したままホイールを手前に回転することで、透明化されたウィンドウが1枚ずつ透明化解除され、前面に表示される。右ボタンを押しながらミドルボタンを押すことで、透明化されたウィンドウがある場合にそのすべてのウィンドウの透明化を解除することができる。さらに、ウィンドウのタイトルバー上で右ボタンを押したままホイールを奥に回転すると、そのウィンドウは表示されたまま、奥行き方向の1つ奥にあるウィンドウを透明化することができる。同様にタイトルバー上で右ボタン

[†] 京都工芸繊維大学 Kyoto Institute Technology

を押したままホイールを手前に回転すると、透明なウィンドウの透明化が解除され、タイトルバーを押しているウィンドウの1つ奥に表示される。

また、提案手法は既存のウィンドウ切り替え手法と共存できるため、例えば、透明化されたウィンドウが存在するときでも、透明化されていないウィンドウであれば他の手法によるウィンドウ切り替えが可能であり、新たなウィンドウを開いたりすることも可能である。透明化を解除する際には解除されたウィンドウが前面に表示される。

4. 実験

提案手法によるウィンドウの切り替えの効率性を調べるため、実験を行った。

4.1 実験タスク

実験に用いたウィンドウレイアウトは、図3のように4層から成り、各層にはタイル状に左右に並べた2枚のウィンドウを配置した。それぞれのウィンドウは幅400ピクセル、高さ500ピクセルで、文章表示あるいは入力領域が1つとボタンが1つずつ配置されている。被験者は、最初に前面のウィンドウAにあるボタンを押し、Aに表示された文章をBに入力する。入力後、B上で指示されたA群内の1つのウィンドウを表示し、さらにこのウィンドウ内で指示されたB群内の1つのウィンドウを表示する。そして、表示されたA群のウィンドウの文章をB群のウィンドウに入力する。さらに、開始時のウィンドウAおよびBを表示し、再度Aの文章をBに入力し、最後にBのボタンを押し、これを、指定されるウィンドウの組み合わせ9通り×2回ずつ、計18回繰り返させた。

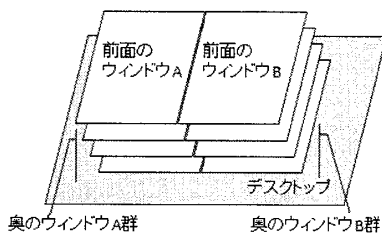


図3：ウィンドウレイアウト

被験者は情報系の大学生・大学院生12名であり、すべての被験者が提案手法、タスクバーを用いた手法、Alt+Tabの3つのウィンドウ切り替え方法を用いて実験を行った。なお、手法についての順序効果が生じないように実験順序のバランスを取った。

それぞれのウィンドウ切り替え方法を比較するために、最初にAのボタンを押してから、最後にBのボタンを押すまでの時間を測定し、文字入力時間を除いた時間をタスク完了時間とした。

4.2 結果

タスク完了時間の結果を図3に示す。なお、システムエラーの生じた1名の被験者のデータが利用できなかったため、11名の被験者の結果について示す。

タスク完了時間において、対応のある一元配置分散分析を行った結果、手法間の主効果が有意であった($F(2,10)=18.459, p<0.05$)。多重比較を行った結果、Alt+Tabよりもタスクバー($p<0.05$)、および提案手法($p<0.05$)が有意にタ

スク完了時間が短い結果となった。しかし、タスクバーと提案手法間には有意差が見られなかった。

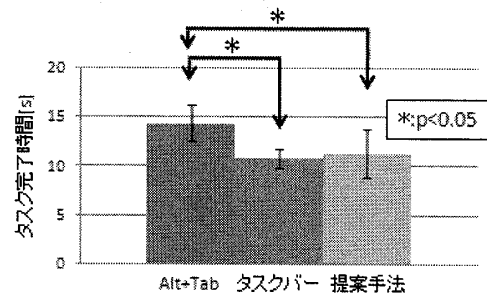


図3：タスク完了時間

5. 考察

タスク完了時間において、提案手法とタスクバーにおいて有意差が見られなかった。各手法についてさらに分析したところ、元のウィンドウへの切り替えに要する時間は提案手法のほうがタスクバーよりも短い、奥にあるウィンドウを参照する際には逆にタスクバーよりも時間がかかっていた。その原因として、タスクバーでは目的のウィンドウを一度で選択できるのに対して、提案手法では奥にあるウィンドウほどホイールの回転が必要となり、操作量が増加するためだと考えられる。また、提案手法では、ホイールを回転させすぎて目的のウィンドウを過ぎてしまった際に逆方向にまた回転させるという手間が生じたことも原因であると考えられる。同様の問題はAlt+Tabでも発生するが、提案手法では元のウィンドウに戻るときの操作量が少ないため、Alt+Tabに比べてウィンドウの切り替えに要する時間が短くなったと考えられる。なお、今回実験に用いたウィンドウレイアウトは、2枚ずつ4層から成るものであり、各層において2枚のウィンドウがタイル状に並び、たがいに重ならないものであった。ウィンドウが互いに重なるような実用的場面では提案手法の有用性がより明確に出る可能性があると考えられる。

6. まとめ

本研究では、PCを用いた作業中に、レイアウトを変更することなくウィンドウを切り替えるためのウィンドウ透明化手法を提案し、その有用性を実験で評価した。実験の結果、提案手法はAlt+Tabよりも有意に早くウィンドウの切り替えができるが、タスクバーを用いた場合との間には有意な差はなかった。今後の課題として、デスクトップへの切り替えや、互いに重なるウィンドウ間の切り替え操作を含む、より実用的なタスクを用いて手法間の違いを評価することが挙げられる。

参考文献

- [1] D.R. Hutchings, G. Smith, B. Meyers, M. Czerwinski, G. Robertson. "Display space usage and window management operation comparisons between single monitor and multiple monitor users", Proc. AVI 2004, 32-39 (2004).
- [2] Quan Xu, G'ery Casiez. "Push-and-Pull Switching: Window Switching based on Window Overlapping", Proc. CHI 2010, 1335-1338 (2010).
- [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, 123-130 (2003).