

リアリティユーザインタフェース -マルチウインドウ環境におけるオブジェクト移動法-

齋藤勉、神場知成、濱川礼、橋本治

NEC

本稿では、複数ウインドウ環境下でのオブジェクト移動方法として、ウインドウ内で表示されている画像をそのまま移動するリアリティ移動法を提案する。これは、従来のGUIにおける直接移動法では、オブジェクトがウインドウ外に出る場合にオブジェクトの表示が変化してしまうため、十分なリアリティ感が維持できないという問題点を解消する。さらに、リアリティ移動法のX-Window上での実現方法を分類、比較検討し、各方法の長所を採り入れた移動ウインドウ準備方法を提案する。評価実験では、本方式は従来方式に較べてウインドウの表示に要する時間が1/2以下で済み、また、従来方式に較べて安定した反応時間が得られた。

Reality User Interface -New object moving method on Multi-Window environment-

Tsutomu Saitoh, Tomonari Kanba, Rei Hamakawa, Osamu Hashimoto

NEC Corporation

1-1, Miyazaki 4-Chome, Miyazaki-ku, Kawasaki, Kanagawa, 216, Japan

In current GUI an image moved by the user is different from the one caught by the user. Thus users find it difficult to regard these two images as the same object and to feel any sense of reality. "Reality Transferring Method", which this paper proposes, is the new method with a sense of reality in transferring information objects between windows on a multi-window environment. Additionally, this paper develops and evaluates its implementation technique called "Preparing Transfer-Window technique". This technique takes less than half the response time of former technique.

1 はじめに

筆者等は現在、リアリティ感のあるユーザインタフェース環境を目標とした、リアリティユーザインタフェース (RUI)[3] を研究開発中である。RUIは、ユーザの現実世界を可能な限り忠実に取り込み、ユーザが違和感や抵抗感を持たずにコンピュータを使えるようにすることを目的としている。

X-Window や Windows などの複数ウィンドウ環境においては、カット & ペーストやアイコンの移動など、ウィンドウ間での情報の受渡しが行われることがある。この情報の移動に良く使われるユーザインタフェース (UI) が直接操作法である。これは、ユーザが情報を表す表示を直接操作して情報の移動を行う手法である。しかし Macintosh や Windows などの従来のシステムでは、この移動の際に使われる表示は簡略化された表現が使われることが多く、ユーザに与えるリアリティ感を損なうことが多かった。

本論文では RUI の観点から、表示されている画像をそのまま移動する「リアリティ移動法」と呼ぶリアリティ感覚のある操作法を提案する。さらにこの手法を X-Window 上で実現する各種方式を比較検討した上で、各手法の長所を採り入れた「移動ウィンドウ準備方式」について述べる。

2 リアリティ移動法の提案

2.1 マルチウィンドウにおけるオブジェクトの移動法

X-Window や Windows などのウィンドウシステムでは、ディスプレイ上に複数のウィンドウを同時に表示することによって、ユーザに多くの情報を同時に提供することができる。ユーザはそれらのウィンドウを渡り歩くように、制御を切替えたり情報を交換したりしながら作業を進めることができる。このような作業環境では、あるウィンドウ中に存在する画像や文字列などを他のウィンドウへ移動することができる。

例えばデスクトップシステムにおいて、作業領域 (ホームディレクトリ) を表す机ウィンドウと格納領域 (他のディレクトリ) を表す本棚ウィンドウがあった場合に、書籍の形で表されたファイルを本棚から机に移して作業をすることができる。以下では、ウィンドウ間で移動される画像や文字列等をオブジェクトと総称する。

このようなオブジェクト移動を行なうユーザインタフェース (以下 UI) には、移動するオブジェクトや移動先の指定、移動の開始にコマンドを用いるコマンド法、メニューを用いて選択して指定するメニュー法、マウスカーソルなどにより、移動するオブジェ

クトや移動先を直接指定する直接操作法がある [1]。

2.2 直接操作法と従来の問題点

マルチウィンドウシステムとして有名な Macintosh や Windows、X.desktop などにおいては、オブジェクトとしてアイコンを用いた直接操作法が採用されている。例えば Macintosh ではアイコンを指定するとその外形が表示され、それを移動する。また Windows ではウィンドウ内ではオブジェクト自体を移動するが、ウィンドウの外に出るとカーソルなどの専用ビットマップイメージを移動する。このように、オブジェクト移動時の表示が固定化されている場合、ビデオ画像や3次元CGを利用したシステムのようにオブジェクトの表示が変化する場合には、移動時のオブジェクトの表示を柔軟に制御することが出来ず、ユーザが実際にオブジェクトを移動しているというリアリティ感が損なわれることが多かった。

2.3 リアリティ移動法の提案

筆者等は現在、リアリティ感のあるユーザインタフェース環境を目標とした、リアリティユーザインタフェース (RUI) を研究開発中である。RUIは、ユーザの現実世界を可能な限り忠実に取り込み、ユーザが違和感や抵抗感を持たずにコンピュータを使えるようにすることを目的としており、画面表示には、ユーザの周りの環境を取り込んだ実写ビデオ映像を使用している。

この RUI 環境下では、オブジェクト移動において単に直接操作法でオブジェクトを移動できるだけでは、ユーザに十分なリアリティ感覚を与えることはできない。そこで本稿では、ウィンドウ内にに表示されているオブジェクトをそのまま別のウィンドウに移動する、直接移動法を提案する。このようなオブジェクトの移動法を、ここではリアリティ移動法と呼ぶ。

具体的な例として、RUIを用いたデスクトップシステムをあげる。ディスプレイ上には、机の上の画像を表示したウィンドウと本棚のウィンドウが存在し、本棚にはオブジェクトとしていくつかの本が収まっているものとする。この本棚の本を机の上に移動する場合、従来の方法では移動中の本は外形や専用のビットマップイメージになってしまうが、リアリティ移動法では本の画像がそのままウィンドウ間を移動することになり (図 1)、ユーザのリアリティ感を損なうことがない。これにより、コンピュータになじみのないユーザに対しても操作時に不安感やストレスを感じさせないユーザインタフェースが作成可能である。

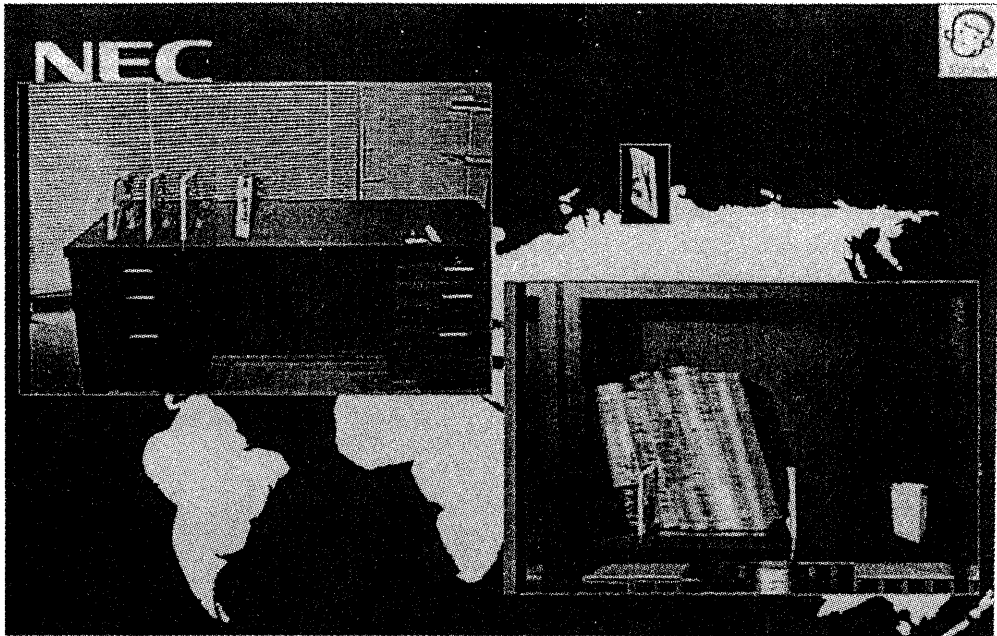


図 1: RUI デスクトップシステムと、オブジェクト (本) のウィンドウ間移動

3 リアリティ移動法の実現方式

3.1 実現方式の検討

一般的なウィンドウシステムにおいては、ある程度複雑な画像表示するためにウィンドウを作成する必要がある。したがって、ウィンドウ内に表示されていたオブジェクト (画像) をウィンドウの外に移動するためには、そのオブジェクトのために別のウィンドウを作成してディスプレイに表示しなくてはならない。この、オブジェクトを移動するためのウィンドウを、ここでは移動ウィンドウと呼ぶ。

この移動ウィンドウの作成および表示をウィンドウ作成の時期で分類すると、以下の3通りが考えられる。

事前作成 初めから全てのオブジェクトを移動ウィンドウとして作成 / 表示しておき、ウィンドウ内部でも別のウィンドウとして扱う

移動開始時作成 ユーザがそのオブジェクトを移動対象として指定した時に作成 / 表示するが、以降はウィンドウ内部でも別のウィンドウとして扱う

脱出時作成 そのオブジェクトがウィンドウの外に移動された時に作成 / 表示する

3.2 実現方式の比較

リアリティ移動方式の実現にあたってはプラットフォームとなるウィンドウシステムに依存するとこ

ろが大きい。ここでは、代表的なウィンドウシステムである X-Window システムを例にとり、上記3通りのタイミングでリアリティ移動方式を実現する場合の利点と欠点を整理した。

3.2.1 事前作成

- (a) 利点 全てのオブジェクトが初めから移動ウィンドウとして表示されているため、ウィンドウ移動時に移動ウィンドウを作成する必要がないので、移動操作に遅れが生じることなく、スムーズに移動できる。
- (b) 欠点 ユーザに対するリアリティ感を増すために、RUI デスクトップではウィンドウの内のオブジェクトについては、そのウィンドウの枠や背景は見えない事が望ましい。これを実現するためには、X-Window システムの shape extension 機能を用いなければならないが、これは重い処理であり、システムの反応が遅くなりユーザに不快感を与える。また、全てのオブジェクトを初めから移動ウィンドウとして作成 / 表示するので、移動しないオブジェクトまで移動ウィンドウにすることになり、無駄な処理が必要になる。

3.2.2 移動開始時作成

- (a) 利点 ウィンドウ脱出時に移動ウィンドウを作成する必要がないので、移動操作に遅れが生じることなく、スムーズに移動できる。

	反応の遅れ	shape extension	無駄な処理
事前作成	(オブジェクト / ウィンドウ作成時)	要	有
移動開始時作成	移動開始時	要	有
脱出時作成	脱出時	不要	無

表 1: リアリティ移動法の各種実現方法の比較

(b) 欠点 オブジェクトを指定した時点で移動ウィンドウを作成 / 表示するため、その時点で反応の遅れが生じる。一般に、オブジェクトの移動を開始する操作は比較的失敗しやすいので、反応の遅れはユーザに強いストレスを与える。また、オブジェクトがウィンドウ内を移動する間も移動ウィンドウとして扱うために、shape extension 問題が生じる。

3.2.3 脱出時作成

(a) 利点 移動対象のオブジェクトを指定した時点では、何もしないので、反応はスムーズであり、また、事前作成をしないので、ウィンドウのオープン時 (表示時) や各オブジェクトの表示時に反応の遅れが生じない。

(b) 欠点 オブジェクトがウィンドウ外に出る時点で移動ウィンドウを作成 / 表示するため、その時点で反応の遅れが生じる。

3.2.4 比較のまとめ

以上の利点 / 欠点を、比較表にまとめた (表 1)。

3.3 移動ウィンドウ準備方式

このように上記の方式はそれぞれ利点 / 欠点を持っており一長一短であるが、以下に提案する移動ウィンドウ準備方式は事前作成 (3.2.1 節) と脱出時作成 (3.2.3 節) を組み合わせたもので、それぞれの利点を活かして欠点をカバーする方式である。

本方式では事前に移動ウィンドウを作成しておくが、移動ウィンドウはオブジェクトが表示されているウィンドウなどの下に隠しておく。ウィンドウ内の移動についてはこの移動ウィンドウを使わずに通常の処理方式でオブジェクト (画像) を移動させる。オブジェクトがウィンドウの外に出るときに、事前に作成してあった移動ウィンドウを下から出して表示する。

本方式では、

- ウィンドウ内の移動には移動ウィンドウではなく元の画像を用いるため、shape extension を用いる必要がなくウィンドウ内の移動がスムーズに行える。

- オブジェクト移動を開始する際の反応に遅れが生じない。
- オブジェクトがウィンドウ外に出る際にも移動ウィンドウの作成という処理がなくなるため、その時点での反応の遅れがなくなる。

という利点がある。

4 移動ウィンドウ準備方式によるリアリティ移動法の実現

4.1 移動ウィンドウ準備方式の手順

移動ウィンドウ準備方式を実現するには以下のような作業が必要となる

1. オブジェクトの画像を移動ウィンドウの画像として指定する。
ここで、RUI デスクトップ [3] では、オブジェクトの画像はウィンドウ内の位置によって変化する。このため、移動ウィンドウの画像はウィンドウから脱出する位置での本の画像が望ましいことから、ウィンドウの縁の部分全てに位置に対応する画像を事前に用意すべきである。しかしこれは効率や資源の観点から不可能なので、オブジェクトの画像から適当に取捨選択することにした。この選択方法については 4.2 節で説明する。
2. 移動ウィンドウを作成しウィンドウの下に隠れるような位置に表示する (図 2 (a))。
3. オブジェクトがウィンドウから外へ出るときオブジェクトの画像に最も近い画像を選び、その画像を持つ移動ウィンドウをウィンドウの背後から前面に出して脱出する位置に表示する (図 2 (b))。

4.2 移動ウィンドウ用画像の選択方法

移動ウィンドウ用画像の選択方法は、ウィンドウ内でのオブジェクト画像の変化の仕方によって様々な方法が考えられるが、今回、オブジェクトの画像はオブジェクトのウィンドウ内での位置に応じて変化するとした。この場合、オブジェクトがウィンドウから脱出する時点での画像はウィンドウの外周付近での画像に限定されるため、移動ウィンドウ用の

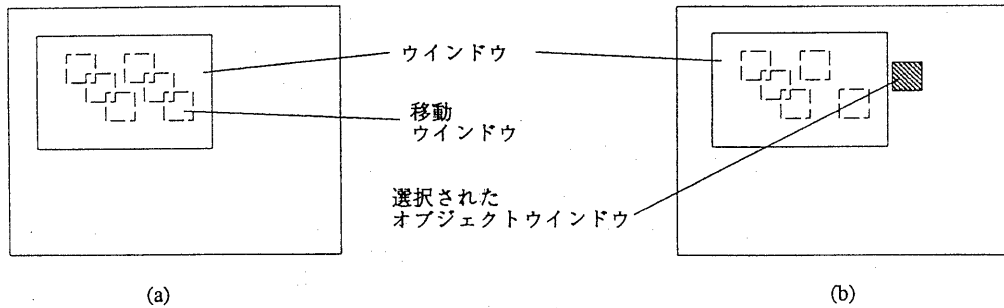


図 2: 移動ウインドウの隠蔽と出現

画像もこの外周付近での画像から適当に間引いて選択すれば良い。

5 評価

5.1 高速性

本方式による脱出時移動の高速性を評価するために、以下の2方式について、プログラムが移動ウインドウの表示指示イベントを受け取ってからウインドウが表示されるまでの時間をそれぞれ10回測定した。

1. 本方式
2. 脱出時作成: 3.2.3 節の方式であり移動ウインドウの作成 / 表示を全て脱出時に行なう。

使用したWSはNEC EWS4800/260(メモリ64MB)、プログラムはInterViews3.0.1および筆者等のグループで開発中のマルチメディア操作用ライブラリXavier[2]上でC++を用い作成した。

測定結果の平均値および最大 / 最小値を表2に示す。

方式	平均値	最小値	最大値
1 (本方式)	7	7	10
2 (脱出時作成)	28	17	43

表 2: 移動ウインドウ表示までの時間 (ms)

評価結果から、本方式(方式1)は2と比較して顕著な時間短縮が見られ、本方式が効果的である事を示している。

方式2.では実際にウインドウ内の画像が表示されるまでの時間に大きなバラツキが見られたが、本方式ではそのようなバラツキは極めて少なく安定した反応時間が保証された。

5.2 移動ウインドウ用画像の違和感

移動ウインドウをあらかじめ用意する本方式の場合、全ての画像に対して移動ウインドウを用意する事はリソースなどの点で不可能である。従って、ウインドウ内でのオブジェクトと移動ウインドウでのオブジェクトの表示に一種のずれが生じ、ユーザが違和感を感じる恐れがある。

そこで移動ウインドウとしてどのようなオブジェクトを用意しておけば、ユーザが違和感なく利用できるかの基準を明確にするため、主観評価実験を行った。

実験用の画像として

- ウインドウ: デスクトップの画像。600x350 pixel
- オブジェクト: 本の画像。本の正面からの角度が-90度から90度まで180度変化する180枚の画像を持ち、ウインドウの横方向の位置によって連続的に表示が切り替わる(左端では-90度、中心では0度、右端では90度)。50x35 pixel (最大)

を用意した。

この本をウインドウの下辺から外に出すときの用意しておく移動用ウインドウ(画像)の個数を1. 3個、2. 5個、3. 10個と変え(図3)、どの程度の違和感があるかを以下の2通りで主観評価を行った。

1. 5段階絶対評価 各ケースについて、5(違和感が気にならない)から1(大いに違和感がある)までの5段階で評価。
2. 3段階相対評価 各辺について10個のケースを基準とし、5個および3個のケースがどれくらい違和感があるかを3(10個のケースとまるで変わらない)から1(大いに違和感がある)までの3段階で評価。

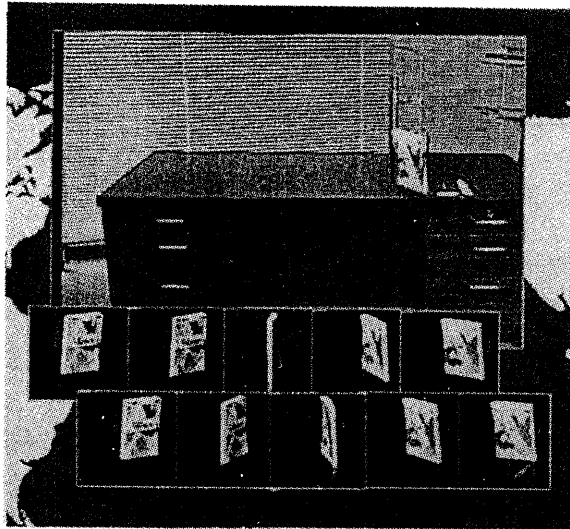


図 3: 主観評価実験用の環境と用意する移動ウインドウ

以上を7人の被験者に評価してもらった結果の平均値を、表3に示す。

移動ウインドウ	10個	5個	3個
5段階絶対評価	4.3	3.7	2.3
3段階相対評価	—	2.6	1.3

表 3: 移動ウインドウの個数による違和感

評価結果から、移動ウインドウを10個用意した場合の違和感は許容できるもので、5個に減らした場合でもそれほど評価は落ちないが、3個ではかなりの違和感が生じることが分かった。

また、被験者から

1. 本の向きが同じなら、多少の角度のずれは気にならない。
2. 本の角度より、本の位置がウインドウの内外でずれた場合に、強い違和感がある。
3. 移動ウインドウの枠が付いていることに対する違和感が大きい。

といったコメントがあった。

1、2については、今後本方式に積極的に採り入れていくつもりである。また3についても、前述のように枠や背景が見えないようにする予定である。

6 おわりに

X-Window や Windows などのマルチウインドウ環境においてオブジェクトをウインドウ間で移動する場合に、表示されている画像をそのまま移動する「リアリティ移動法」を提案し、さらにこれを X-Window 上で実現する方法として「移動ウインドウ準備方式」を開発・評価し、従来方法と比較して効果があることを確認した。

今後は、実用化に向けて、実験で得られた評価およびコメントに基づき、

- オブジェクトの出現位置の高精度化
- 移動ウインドウの表示タイミングの検討
- 移動ウインドウの枠の消去

などについて本手法の改善を進めるとともに、さらに本手法も含めた RUI の研究開発を進めて行く予定である。

参考文献

- [1] Ben Shneiderman. *Designing the User Interface 2nd edition*. Addison-Wesley Publishing Co. Inc., 1992.
- [2] 坂上, 濱川, 暦本. HyperStation: オブジェクト指向 GUI ツール InterViews の AV 拡張. 45 情報処大全, 1992. 5B-01.
- [3] 神場知成, 橋本治. リアリティユーザインタフェースにおけるビデオ映像インタラクション. 47 情報処大全, pp. 185-186, 1993. 2J-2.