

# デスクトップ上の画面変化に基づく 取り消し操作の可視化機構の設計と実装

坂本 有沙<sup>1</sup> 片山 拓也<sup>1</sup> 寺田 努<sup>1,2</sup> 塚本 昌彦<sup>1</sup>

**概要:** GUIにおいてユーザの直前の作業内容を取り消す操作は、あらゆるアプリケーションに導入されており、広く普及している。しかし、従来の取り消し操作は、瞬時に操作が実行されるため取り消された部分を見逃したり、作業を一時的に中断した場合や複数の作業を並行して行っている場合、取り消し操作が実行される部分がディスプレイ領域外に存在する場合にユーザが取り消された内容を把握できない状況が生じる。これまでも取り消し操作を支援する研究が行われているが、アプリケーションに依存し、汎用的なものは存在しない。そこで本研究では、アプリケーションに依存しない取り消し操作の可視化手法を提案する。提案手法では、取り消し操作実行時に視覚的变化が起こることに着目し、デスクトップ上の画面変化を検出することで、取り消し操作の対象を特定する。本稿では、取り消し範囲と取り消し内容、ディスプレイの表示領域外での取り消し操作の明確化のために強調表示機能と取り消し内容提示機能、取り消し通知機能を実装し、その問題点と改良案を考察する。

**キーワード:** 取り消し操作, 可視化, 画面変化

## Design and Implementation of a Mechanism for Visualizing Undo Operations Based on Changes of Desktop Screen

ARISA SAKAMOTO<sup>1</sup> TAKUYA KATAYAMA<sup>1</sup> TSUTOMU TERADA<sup>1,2</sup> MASAHICO TSUKAMOTO<sup>1</sup>

**Abstract:** Undo operation in Graphical User Interfaces is implemented on most applications and widely used. However, conventional implementation of undo operations has the following three problems: (1) users may miss the undone part because the operation is instantaneously worked, (2) it is difficult to know the undone contents if they temporarily interrupt a work or they work multiple operations in parallel, and (3) it is difficult to know the undone contents if the undone part stays outside the visible area of desktop. Though there have been several researches to improve undo operations, they are specialized to a certain application. In this research, we propose a visualization mechanism for undo operation, which is independent of running applications. Our proposed method specifies the area for undo operation by detecting changes of desktop screen. We have implemented several visualization methods including emphasizing the undo area, undone contents presentation, and notification that undone part is outside the visible area.

**Keywords:** undo operation, visualization, changes in screen

### 1. はじめに

パーソナルコンピュータで利用されている GUI (Graphical User Interface) では、一般に取り消し操作によって簡単に直前の作業を取り消せる。しかし、この取り消し操作

<sup>1</sup> 神戸大学大学院工学研究科  
Graduate School of Engineering, Kobe University  
<sup>2</sup> 科学技術振興機構さきがけ  
PRESTO, Japan Science and Technology Agency

は瞬時に起こるため、取り消された対象の位置が分からない、あるいは位置が分かっても取り消された内容が把握できないという問題点がある。この問題に対し、Microsoft社のOffice[1]では、「元に戻す」ボタン横のプルダウンメニューを展開すれば、取り消せる対象が履歴として提示される。またAdobe社Photoshop[2]などのソフトウェアでは、操作履歴を常時視認できるヒストリウィンドウを備える。これらの機能により、ユーザは取り消される対象を取り消し操作実行前に確認でき、任意の操作までのすべての操作が一括で取り消せる。しかし、これらの機能は、ソフトウェアごとに提示方法が異なるためユーザを困惑させたり、「入力」や「フォントの色」といった操作の表現が簡略化されているため、取り消される作業内容を正確に把握するのが難しい。また、ユーザの意図した取り消し操作が行われなかった場合に变化した箇所を見逃す、取り消し部分がディスプレイ領域外に存在する場合に取り消し操作が実行されたとしても気付かないという問題点も存在する。

上記で述べた問題点以外にも、任意の操作のみの取り消しができないという問題や、Google Document[3]のように複数のユーザによって編集されるドキュメントに対して、他ユーザの編集の取り消しが行えないという問題点がある。これらの問題点を解決するために、選択的な取り消しを実現した研究[4]や取り消し操作にグラフィカルな操作履歴を用いた研究[5]、共同編集可能な環境での取り消し操作を可能にする研究[6]など、様々な研究が行われている。しかし、これらは専用のアプリケーションに依存した設計になっており、汎用的に用いることができない。

そこで本研究では、取り消し操作実行時には一般にデスクトップ上に視覚的变化が起こるといった特徴をもとに、デスクトップの画像変化を用いてアプリケーションに依存しない取り消し操作の可視化手法を提案する。提案手法は、どこが修正・変更されたかが分かりにくい、取り消された操作の詳細が分からない、ディスプレイ領域外での取り消し操作に気付かない、という3つの問題点を解決するため、取り消された点を強調表示する機能、取り消された入力系列を表示する機能、画面外での取り消しを通知する機能を備える。

本稿は以下のように構成されている。2章では関連研究について述べ、3章で取り消し操作に伴う画像変化について論じる。4章では提案手法について述べ、5章ではシステムの実装について述べ、提案手法の問題点と改良案について考察し、最後に6章で本稿のまとめを行う。

## 2. 関連研究

通常を取り消し操作において、Microsoft社Officeなど多くのアプリケーションでは過去の操作を順に遡って取り消しを実行するため、途中の一部の操作を選択的に取り消せない。そこで、Adobe社Photoshopにおいて一部実現

されているように任意の操作のみの取り消しを可能にしたのが選択的取り消しである。選択的取り消しを用いることで、ユーザはそれまでの操作の実行順序にとらわれず、任意の操作を取り消せる[7]。TRIBASEと呼ばれるプラットフォームで実装された選択的取り消しでは、選択した対象が既に存在せず取り消し操作を行えない場合をインアクティブであるとし、反対に対象が存在して取り消し操作を行える場合をアクティブであると定義することで、その取り消し対象が取り消し可能かを判断し、これまでの操作の関係性を壊さないように配慮している[8]。また、インタラクティブなGUIの開発環境であるAmulet[9]では、コマンドオブジェクトを階層化することで、操作を選択的に取り消せるだけでなく、その操作を再利用し同じオブジェクトに対して再実行したり、他のオブジェクトに対して新しく実行できる[10]。

この他にも、テキストエディタや表計算ソフトなどに限定されるが、ユーザが選択した領域内で行われた操作のみを取り消すというRegional Undo[11]と呼ばれる手法がある。この手法では、取り消しを適用したい領域を選択できるため、ユーザが想定している対象に対して取り消し操作を行える。

Microsoft社OfficeやAdobe社Photoshopにおいて取り消し可能な操作履歴が確認できることを述べたが、これらの履歴は操作の表現が簡略化されているため、ユーザが望むコマンドや状態を見つけるのは困難である。この問題点を解決するため、操作履歴のグラフィック表示やオブジェクトごとの履歴保存を用いて選択的取り消しを実現する手法が提案されている。西田らは、ペイントやブラウザにおいて、取り消し操作を「過去の状態を回復させる操作」ととらえて、すべての過去の作業状態の遷移をわかりやすく提示するためにヒストリグラフを用いている[12]。グラフのノードは作業状態を示し、それに続くノードとその間を結ぶエッジは操作を表し、ノードを選択するとそのノードと対応する作業状態を回復できる。増田らは、選択的取り消しを容易かつ有効に利用するためのインタフェースとして、専用のグラフィカル履歴ブラウザを設計している[13]。この履歴ブラウザでは図形オブジェクトごとに操作コマンドをグループ化し、作業画面のスナップショットから成るグラフィカル履歴をフリップコミックで提供している[14]。取り消したい操作の描かれているスナップショットを選択するだけで希望の操作が取り消される。この時、ユーザは操作の内容や効果を気にすることなく、視覚的に対象のオブジェクトの状態を理解できる。また、コマンドをグループ化したことで、特定のオブジェクトを選択するとそのオブジェクトに関連する作業内容のみを提示するので、取り消したい操作を検出する手間も減少する。この他にもFlatland[15][16]と呼ばれるシステムをもとに拡張されたホワイトボードに線や図形などオブジェクトご

との履歴を保存することで、特定のオブジェクトに対する操作のみを対象とした取り消し操作や、複数オブジェクトを対象とした操作の取り消しにも対応したシステムが提案されている [17] .

複数のユーザの共同作業においては、操作を行っていないユーザによって取り消し操作が行われると一貫性が崩れてしまうので、適切な修正を加える必要がある。そこで、Prakash らは、非線形の取り消し操作が可能な DistEdit ツールキットにおける US&R モデル [18] を用いて、取り消しを行わない部分は現在の状態を保ったまま、取り消しを行う部分のみを取り消すことで、他のユーザの操作によって生じた作業状態を考慮しながら個人的に操作した部分を取り消すことができる手法 [19] を提案している。また、Weiss らの手法 [20] は、Compensation と呼ばれるフレームワークを用いて複数ユーザの操作の一貫性を維持しながら、共有ドキュメントでの取り消し操作を実行できる。

1 章で述べた問題点を解決するために上記の研究が行われているが、これらは専用のアプリケーションに依存した設計である。そこで、汎用的に用いることのできるシステムの提案が必要となる。

### 3. 取り消し操作に伴う画像変化

本研究では、取り消し操作に伴ってデスクトップ上に視覚的变化が起こることに着目し、デスクトップの画像変化を利用した汎用的な取り消し操作の可視化手法を提案する。提案手法の設計にあたり、以下の 9 つのアプリケーションにおける取り消し操作の挙動を調査した。今回の調査には、Windows7 搭載の Lenovo ThinkPad X220 (CPU: Core i5 2.50GHz, メモリ: 4.00GB) を用いた。

- Microsoft Office PowerPoint 2007 (以下, PowerPoint)
- Microsoft Office Word 2007 (以下, Word)
- Microsoft Office Excel 2007 (以下, Excel)
- ペイント
- エクスプローラ
- メモ帳
- Mozilla Firefox ver. 9.0.1 (以下, Firefox)
- Mozilla Thunderbird (以下, Thunderbird)
- Microsoft Visual Studio 2008 (以下, Visual Studio)

各アプリケーションにおいて取り消し操作が可能な操作の一部を表 1 に示す。常に視覚的变化があるものを  $\square$  で示し、ユーザが視覚的变化を確認するために条件があるものを  $\square$  で示す。

表 1 の操作の種類において、テキスト編集はテキストの入力や修正、切り取り、貼り付けを示し、図形編集は図形の移動や削除、複製、挿入、位置やサイズの変更、回転を示す。どのアプリケーションにおいても、テキスト編集や図形編集の他に、コピー・切り取り・貼り付けといった汎用的な機能など、ほぼ同様の操作を取り消し対象にしてい

表 1 各アプリケーションにおける取り消し可能な操作  
Table 1 Undoable operations for each application

アプリケーション	操作の種類	視覚的变化
PowerPoint	デザイン変更 図形編集 レイヤ変更 スライド編集 ページ設定 アニメーション変更 テキスト編集	
Word	インデント変更 行間変更 テキスト編集 図形編集 レイヤ変更 箇条書きの変更 段落番号の編集 フォント変更 ワードアート変更 配置変更	
Excel	セル内テキスト編集 セル編集 グラフ編集 レイヤ変更	
ペイント	線の描画 塗りつぶし 消しゴムでの消去 図形編集	
エクスプローラ	ファイル管理	
メモ帳	テキスト編集	
Firefox	テキスト編集 Google カレンダー編集 ブックマーク管理	
Thunderbird	テキスト編集 メール移動 メール削除	
Visual Studio	テキスト編集 ツール編集	

ることが分かる。しかし、PowerPoint や Word や Excel におけるレイヤ変更では、ユーザが視覚的变化を確認するための条件が存在する。例えば、他のオブジェクトと重なっていないオブジェクトに対して表示レイヤの設定を行う場合や、より前面にあるオブジェクトと重なっていないオブジェクトのレイヤを最前面に切り替える場合は視覚的变化を確認できない。また、通常は視覚的变化を伴う操作であっても、取り消し操作を実行する部分がディスプレイの領域外に存在する場合も視覚的变化を確認できない場合もある。例えば、対象のウィンドウの一部がディスプレイの領域外に存在する場合、PowerPoint, Word, Excel, ペイントにおいては画面上部のツールバーもしくはタイトルバーに視覚的变化が発生する場合があるが、それ以外のア

アプリケーションでは視覚的变化が発生するにも関わらず、それがディスプレイ領域外のため、ユーザは視覚的变化を確認できない。また、スクロールバーを伴う広範囲にわたるコンテンツにおいて、非表示領域で取り消し操作が行われる場合、PowerPoint や Word, Excel, メモ帳, Visual Studio では、取り消し操作と同時に取り消し部分が確認できるよう自動でウィンドウの表示領域に移動する。それに対して、ペイントや Thunderbird, エクスプローラでは取り消し操作は行われるが表示領域には移動しない。特に、エクスプローラでは、取り消し操作が行われる際、操作の確認を促す別ウィンドウが出現するため、取り消し操作があったことが確認できる。ユーザが取り消し操作に伴う視覚的变化を確認できない場合、取り消された対象が把握できないだけでなく、取り消し操作が行われたかどうか把握できない場合がある。さらに、ユーザが取り消し操作が行われていないと勘違いをしてしまい、複数回の取り消し操作を実行してしまう問題も発生する。そのため、視覚的に変化があるが、ユーザがそれを確認できない場合でも取り消し操作が行われたことをユーザに知らせる機構が必要になる。

以上の調査結果より、大部分の取り消し操作は視覚的变化を伴うことから、画像差分を用いた取り消し操作の認識によって、アプリケーションやプラットフォームに依存しない、汎用的な取り消し操作の機能拡張が行えると考えられる。一方、画面外にウィンドウが存在する場合など単純に画像差分を用いるだけではうまくいかない場合が存在するため、そのような場合でも取り消し内容を把握できる機能を実現する必要がある。

#### 4. 提案手法

本研究では、取り消し操作に伴ってウィンドウに視覚的变化があることを利用し、どこが修正・変更されたか分かりにくい、取り消された操作の詳細が分からない、ディスプレイの領域外での取り消し操作に気付かないという従来の取り消し操作の問題点に対して、以下の3つの機能を提案する。

- 強調表示機能: ユーザの取り消し操作によって変化した部分に視覚的效果を与えて表示
- 取り消し内容提示機能: 取り消し操作によって取り消された入力内容をユーザに提示
- 取り消し通知機能: ディスプレイ領域外で取り消し操作が起こったことをユーザに通知

その際、本研究では取り消し操作のトリガとして、取り消しによく用いられているショートカットである Ctrl キーと Z キーの同時押下を用いる。ここで、すべての提案手法において取り消し操作を検知するために提案システムはキーボードの入力のフックを行い、Ctrl キーと Z キーが同時に押下された際に、上記の3種類の機能を使って取り



図 1 赤枠表示機能

Fig. 1 Function of showing a red rectangle

消し操作を可視化する。以下、それぞれの機能の詳細を述べる。

##### 4.1 強調表示機能

取り消し部分を明確にするための強調表示機能として、取り消し部分を赤い枠で強調する赤枠表示機能と、取り消し操作を滑らかに提示する滑らか表示機能を実装した。これらの機能によって、取り消し操作によって変化した箇所の特長を容易にすると同時に、取り消される作業内容をユーザが把握しやすくなる。

##### 赤枠表示機能

提案手法では、Ctrl キーと Z キーが同時に押下された瞬間のスクリーンショットを取り消し操作前のディスプレイ状態として取得し、その 0.5 秒後のスクリーンショットを取り消し操作後のものとして取得する。次にそれぞれのスクリーンショットにおける各ピクセルの RGB 値を比較し、RGB 値が異なるピクセルの座標を記録する。ここで、本研究では、高速化のために RGB 値のうち B (青) の値のみを比較している。これらの座標の全てを囲む最小の四角形を取り消し操作が実行された範囲として設定し赤枠を描画すると図 1 に示すように赤枠が表示され、ユーザに対し取り消し部分を強調して表示できる。

##### 滑らか表示機能

赤枠表示機能同様、Ctrl キーと Z キーが同時に押下された瞬間のスクリーンショットを取り消し操作前のディスプレイ状態として取得し、そのスクリーンショットをディスプレイに重ねて描画する。その描画したスクリーンショットの不透明度を設定することで、次第に取り消し操作実行後のディスプレイが見えるようになる。ここでは不透明度の初期値を 100% に設定した状態から 30 ミリ秒間隔で 1% ずつ減少させていくことで、最終的に不透明度を 0% にし、重ねたスクリーンショットが見えない状態にしている。これによって、通常では図 2(a) に示すように実行される取り消し操作が、図 2(b) に示すように瞬時に実行された取り消し操作をまるで徐々に実行されているかのようにユーザに提示できる。ここで、左のスクリーンショットが取り消し操作を行う前の状態を表し、右のスクリーンショットが滑らか表示機能が完了した状態、つまり取り消し操作後の状態を表す。

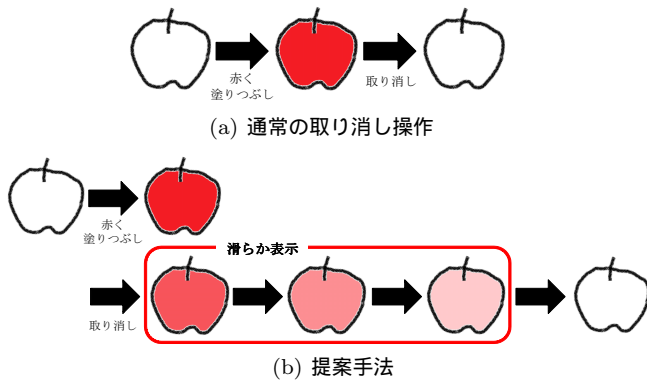


図 2 滑らか表示機能

Fig. 2 Function of showing trajectory smoothly

#### 4.2 取り消し内容提示機能

ユーザに取り消し内容を提示するために、システムフックを用いて逐次記録した作業内容から取り消し操作によって取り消された入力内容を把握する。その把握内容を用いて、以下の問題を解決するための付加機能を実現する。

従来の取り消し操作は、コンテンツの変化のみに対応している。そのため、直接コンテンツに影響を与えない変化、例えばペイントにおける線の色・太さの選択、文書作成におけるフォントやサイズ・効果の選択などには非対応で、再び同じ状態にすることが難しい場合がある。そこで、把握したキーボードとマウスの操作履歴と画像差分を用いて、取り消し操作によって戻された期間を特定し、その期間のキーボードとマウスの入力内容をユーザに提示することでユーザは過去に行った操作を再現できる。

##### 入力の記録

本機能において提示する内容は、キーボードとマウスの入力である。提案システムはそれぞれの入力イベントをフックし、1 秒間隔でそれぞれに対応するスクリーンショットと関連付けて保存する。ここで、記録するキーボードとマウスの以下の入力イベントはそれぞれ以下の通りである。

- キーボード: 各キーの KeyDown イベント
- マウス: MouseDown イベント, MouseUp イベント, MouseMove イベント

なお、マウスの入力はこの 3 種類のイベントを組み合わせることで通常のカーソル移動とドラッグを識別できる。

##### 期間の特定

提案手法は、取り消し操作後のスクリーンショットとそれ以前に定期的に取得されたスクリーンショットを比較し、類似したスクリーンショットを探索し、取り消された期間を特定する。このスクリーンショットの比較では、画像の RGB 値で正規化相互相関係数を求めるテンプレートマッチングを用いる [21][22]。以降で述べる計算式を用いて、しきい値以上の数値が得られた場合、比較対象のスクリーン

ショットが取得された時刻以降の操作が取り消されたと定義する。具体的には、個別に格納したスクリーンショットを、その取得時間の新しい順に取り出し、その RGB 値を取得する。ここで、スクリーンショットの  $x$  座標を  $m$ 、 $y$  座標を  $n$  とすると、それ以前のスクリーンショットの  $(m, n)$  における RGB 値を  $f(m, n)$ 、取り消し操作後  $g(m, n)$  とできる。さらに、それぞれの RGB 値の平均値を  $\bar{f}$ 、 $\bar{g}$  として、

$$F = f(m, n) - \bar{f} \quad (1)$$

$$G = g(m, n) - \bar{g} \quad (2)$$

とおくと、正規化相互相関  $R$  の計算式は、

$$R = \frac{\sum_{n=1}^N \sum_{m=1}^M FG}{\sqrt{(\sum_{n=1}^N \sum_{m=1}^M F^2)} \sqrt{(\sum_{n=1}^N \sum_{m=1}^M G^2)}} \quad (3)$$

で表される。 $R$  は最大で 1 となり、この時比較する二つのスクリーンショットが 100% 一致していることを表す。しかし、操作対象が全く同じ状態であったとしても、その他の環境、例えばツールバーやステータスバーの時計、バッテリー状態などの変化が比較に影響を与える可能性が考えられる。そこで、この状況を考慮し、提案システムでは予備実験の結果をもとに  $R$  の値に以下の条件を設けた。ペイントなど画像編集を主に行うアプリケーションについては視覚的変化の大きい取り消し操作が多いので、 $R > 0.98$  の時に二つのスクリーンショットが一致しているものとして扱い、メモ帳などテキスト編集を主に行うアプリケーションについては視覚的変化の小さい取り消し操作が多いと考えられるので、 $R > 0.99$  の時に二つのスクリーンショットが一致しているものとして扱う。ここでのアプリケーションの区別は、現段階では各スクリーンショットに関連付けられたキーボードの入力イベントが 3 を超えるものが存在する場合に  $R > 0.99$  を適用し、それ以外の場合は  $R > 0.98$  を適用する。ここで実行例として、オレンジの実線で囲んだ部分の 3 本の黒い実線を消しゴムで消す作業を取り消した様子を図 3 に示す。ここで、図 3 のキャプションは取り消し操作の何秒前かを表す。取り消し操作後のスクリーンショット (図 3(a)) と取り消し操作前の各スクリーンショット (図 3(b), 図 3(c), 図 3(d)) との正規化相互相関係数の値  $R$  を表 2 に示す。図 3(d) のスクリーンショット比較時に、 $R$  がしきい値を超えたため、取り消し操作によって 10 秒前まで戻ったと判断できる。

##### 内容の提示

上記の時間の特定方法を用いて得られた取り消し操作によって戻された時間をもとに、取り消し内容の提示を行う。この時間以降に記録されたスクリーンショットに関連付けられているキーボードとマウスの入力履歴を結合し、得ら

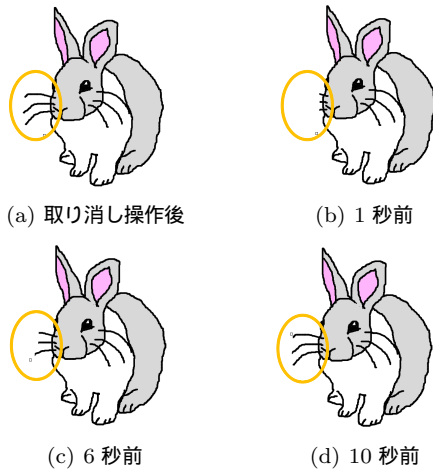


図 3 スクリーンショットの比較  
Fig. 3 Comparison of screenshots

表 2 各スクリーンショットの正規化相互相関係数

Table 2 Normalized cross-correlation coefficient for each screenshot

スクリーンショット	R
取り消し操作 1 秒前 (図 3(b))	0.980
取り消し操作 6 秒前 (図 3(c))	0.988
取り消し操作 10 秒前 (図 3(d))	0.997

れた一連の入力履歴を図 4 や図 5 のように提示する。ここで、図 4 における例はキーボードの入力イベント履歴を提示するもの、図 5 はマウスの入力イベント履歴を提示し、マウスの軌跡を線で提示しているものである。マウスの軌跡は単なるカーソル移動をグレーの点線、ドラッグを黒い実線で示し、マウスボタンのイベントがあった点を青く表示している。図 4 の場合、オレンジで囲んだ部分においてアルファベットキーと変換の Space キー、確定の Enter キーが入力されたことがわかる。また、図 5 の場合、マウス操作として色選択部分からの移動と線の描画開始時と終了時のクリック、描画のためのドラッグ操作があったことがわかる。

### 4.3 取り消し通知機能

3 章で述べたように、従来の取り消し操作において、ディスプレイ領域外で取り消し操作が行われた場合、操作によるデスクトップ画面の変更を確認できない。また、一部のアプリケーションにおけるツールバーやクイックアクセスツールバーのグラフィックの変化により取り消し操作の実行を確認できるが、その内容については確認できない。そこで、ディスプレイ領域外で取り消し操作が実行された場合のみユーザに通知し、ユーザが任意に取り消された内容を取得できる機能を提案する。本稿では、ウィンドウの点滅により取り消し操作実行を通知することとした。具体的には、Ctrl キーと Z キーが同時に押下された瞬間にアク

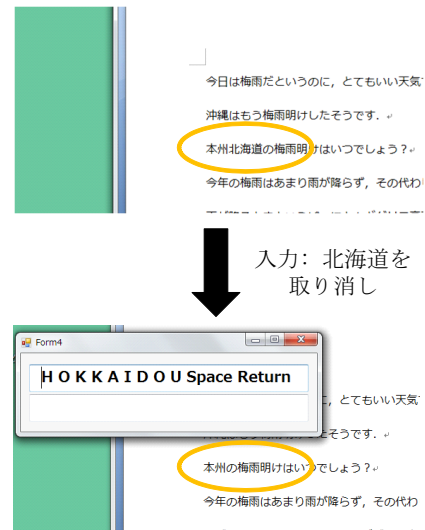


図 4 取り消し内容提示機能: キーイベント

Fig. 4 Function of showing undo contents: Keyboard event



図 5 取り消し内容提示機能: マウスイベント

Fig. 5 Function of showing undo contents: Mouse event

ティブなウィンドウのハンドルを取得し、そのハンドルをもつウィンドウに対して画面外での取り消し操作が行われたことを検出するとタスクバーボタンおよびタイトルバーを点滅させる。

ディスプレイ領域外のウィンドウをキャプチャするために、非表示領域のスクリーンショットを取得できる Print-Window 関数を用いた。具体的には、アクティブになったウィンドウのスクリーンショットを、アプリケーションご

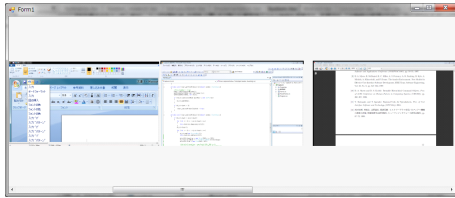


図 6 ウィンドウ追跡の例

Fig. 6 A screenshot of window tracking function

とに記録し続ける．ある時間における追跡の様子を図 6 に示す．ここでは，左からペイント，Visual Studio，PDF ビューアをキャプチャしている．提案システムは，取り消し操作が行われた際にアクティブなウィンドウのスクリーンショットとその履歴を順次比較し，取り消された位置と内容を特定する．その際，取り消された位置がディスプレイの領域外にある際にはウィンドウを点滅させている．

## 5. 実装と考察

提案システムのプロトタイプを実装した．実装には OS として Microsoft Windows7 搭載の Lenovo ThinkPad X220 (CPU: Core i5 2.50GHz, メモリ: 4.00GB) を用い，Microsoft Visual C#2008 により実装した．以下，プロトタイプを利用した結果から提案手法を考察する．

### 5.1 取り消し操作に対する挙動の差

取り消し操作に対する挙動はアプリケーションごとに異なる．例えば，取り消し操作を連続的に行った場合，多くのアプリケーションでは取り消し可能な操作が存在しなくなるまで取り消し操作を行い，それ以降は何も起こらないが，Excel でのセル内のテキスト編集や，メモ帳におけるテキスト編集では，直前の操作のみ取り消しが可能であり，連続的に取り消すことはできない．提案システムでは，ユーザの入力履歴とスクリーンショットの変化を合わせて保持しているため，ユーザの入力を遡らせるような入力をエミュレートすることで，従来の取り消し操作では取り消せなかった範囲の取り消し操作を実行できる可能性がある．

### 5.2 取り消し操作の入力方法

現在の実装ではキーボード入力による取り消し操作にしか対応していないため，多くのアプリケーションに標準的に内蔵されているメニューバーでの取り消し操作や PowerPoint や Word，Excel，ペイントなど特定のアプリケーションに設けられているクイックアクセスツールバーでの取り消し操作へ対応できない．今後は，これらの入力による取り消し操作への対応を検討している．

### 5.3 可視化方法の改善

赤枠表示機能において，ユーザの操作以外の画面変化，例えば，Microsoft 社 Office におけるツールバーでの変化

や PowerPoint における全スライドの縮小表示内の変化などもユーザの操作による変化と誤認して，実際の取り消し操作が実行された部分と異なる範囲に赤枠を表示してしまうことがある．また，複数の点在した部分への取り消し操作に対する個別の強調表示も現状では対応していない．これらの問題点の解決方法を検討し，より精度の高い取り消し部分の特定を目指す．

取り消し内容提示機能において，記録したキーボード入力やマウス操作の情報をそのまま提示するのではなく，ユーザが容易に理解できる自然な情報提示方法の検討や，精度の高い情報提示のためのスクリーンショットやキーボード入力・マウス操作を取得する最適な時間間隔の検証を行う．また，類似のスクリーンショット探索での正規化相互相関の数値およびアプリケーションの区別方法は現在も検証段階であり，より精度の高いマッチングの実装を今後の課題とする．

取り消し通知機能において，ユーザに直観的に取り消された位置を提示する方法や，理解しやすい取り消し内容の提示方法を検討する．ディスプレイの領域外で取り消し操作が行われた場合，対象が表示されていないため，よりユーザの理解を助ける内容の提示方法が必要となる．

以上の機能の改良の後，評価実験を行い，提案手法が有効であるかの検証・考察を行う予定である．

## 6. おわりに

本研究では，デスクトップ上の画面変化とユーザインタフェースの操作履歴を用いることで，アプリケーションに依存せず，取り消し操作を可視化するシステムを提案した．本稿では，どこが修正・変更されたか分かりにくい，取り消された操作の詳細が分からない，画面外の取り消し操作に気付かないという問題点に対して，強調表示機能，取り消し内容提示機能，取り消し通知機能の実装を行った．今後は，赤枠表示機能において取り消し部分の特定精度を高め，複数の点在した部分への取り消し操作に対する個別の強調表示の方法を検討する．取り消し内容提示機能において，ユーザへのより自然な情報提示方法や適度な輝度値の条件を検討する．取り消し通知機能において，通知方法を検討し，評価実験を行う予定である．これらを改善することで，取り消し操作における問題点を解決し，ユーザの操作をサポートする，より快適な取り消し操作を提供できる．

謝辞 本研究の一部は，科学技術振興機構戦略的創造研究推進事業（さきがけ）および文部科学省科学研究費補助金基盤研究（A）(20240009) によるものである．ここに記して謝意を表す．

## 参考文献

- [1] Microsoft Office, <http://office.microsoft.com/>.
- [2] Adobe Photoshop,

- <http://www.adobe.com/photoshop/>.
- [3] Google, <http://www.google.co.jp/>.
- [4] A. G. Cass and C. S. T. Fernandes: Using Task Models for Cascading Selective Undo, *Proc. of the 5th International Workshop on Task Models and Diagrams for Users Interface Design (TAMODIA2006)*, pp. 186–201 (2006).
- [5] D. Kurlander and S. Feiner: A Visual Language for Browsing, Undoing, and Redoing Graphical Interface Commands, *Visual Languages and Visual Programming*, Plenum Press, pp. 257–275 (1990).
- [6] D. Chen and C. Sun: Undoing Any Operation in Collaborative Graphics Editing, *Proc. of the 2001 International ACM SIGGROUP Conference on Supporting Group Work (GROUP2001)*, pp. 197–206 (2001).
- [7] T. Beriage: A Selective Undo Mechanism for Graphical User Interfaces Based On Command Objects, *ACM Transaction on Computer-Human Interaction*, Vol. 1, No. 3, pp. 269–294 (1994).
- [8] C. Zhou and A. Imamiya: Object-based Nonlinear Undo Model, *Proc. of the 21th Computer Software and Applications Conference (COMPSAC1997)*, pp. 50–55 (1997).
- [9] B. A. Myers, R. McDaniel, R. C. Miller, A. S. Ferreny, A. D. Faulring, B. Kyle, A. Mickish, A. Klimovitski, and P. Doans: The Amulet Environment: New Models for Effective User Interface Software Development, *IEEE Transaction on Software Engineering*, Vol. 23, No. 6, pp. 347–364 (1997).
- [10] B. A. Myers and D. S. Kosbie: Reusable Hierarchical Command Objects, *Proc. of SIGCHI Conference on Human Factors in Computing Systems (CHI1996)*, pp. 260–267 (1996).
- [11] Y. Kawasaki and T. Igarashi: Regional Undo for Spreadsheets, *Proc. of the 17th Annual ACM Symposium on User Interface Software and Technology (UIST2004)*, Demo abstract (2004).
- [12] 西田知博, 林 真志, 辻野嘉宏, 都倉信樹: ヒストリーグラフを用いたアンドゥ機構の提案と評価, 情報処理学会研究報告, ヒューマンインタフェース研究会報告, Vol. 99, No. 69, pp. 67–72 (1999).
- [13] 増田尚則, 今宮淳美: Undo 機能をもつグラフィカル履歴ブラウザ設計と視覚的探索方法, 電子情報通信学会論文誌, Vol. J85-D1, No. 8, pp. 798–810 (2002).
- [14] C. Meng, M. Yasue, A. Imamiya, and X. Mao: Visualizing Histories for Selective Undo and Redo, *Proc. of the 3rd Asia Pacific Computer and Human Interaction (APCHI1998)*, pp. 459–464 (1998).
- [15] T. Igarashi, W. K. Edwards, A. LaMarca, and E. D. Mynatt: An Architecture for Pen-based Interaction on Electronic Whiteboards, *Proc. of International Working Conference on Advanced Visual Interfaces (AVI2000)*, pp. 68–75 (2000).
- [16] E. D. Mynatt, T. Igarashi, W. K. Edwards, and A. LaMarca: Flatland: New Dimensions in Office Whiteboards, *Proc. of SIGCHI Conference on Human Factors in Computing Systems (CHI1999)*, pp. 346–353 (1999).
- [17] W. K. Edwards, T. Igarashi, A. LaMarca, and E. D. Mynatt: A Temporal Model for Multi-Level Undo and Redo, *Proc. of the 13th Annual ACM Symposium on User Interface Software and Technology (UIST2000)*, pp. 31–40 (2000).
- [18] J. S. Vitter: US&R: A New Framework for Redoing, *Journal of IEEE Software*, Vol. 1, No. 4, pp. 39–52 (1984).
- [19] A. Prakash and M. J. Knister: A Framework for Undoing Actions in Collaborative Systems, *ACM Transaction on Computer-Human Interaction (TOCHI)*, Vol. 1, No. 4, pp. 295–330 (1994).
- [20] S. Weiss, P. Urso, and P. Molli: A Flexible Undo Framework for Collaborative Editing, *Technical Report of Institut National de Recherche en Informatique (INRIA)*, No. 6516 (2008).
- [21] 池田光二, 吉田昌司, 中島啓介, 浜田長張晴, 依田晴夫: 正規化相関演算の単調関数化による高速テンプレートマッチング, 電子情報通信学会論文誌, Vol. J83-D2, No. 9, pp. 1861–1869 (2000).
- [22] 金谷健一, 金澤 靖: テンプレートマッチングによる対応探索の自動しきい値設定法, 電子情報通信学会論文誌, Vol. J86-A, No. 12, pp. 1502–1509 (2003).