

## Rewind-ow: 特定領域の時間を巻き戻して閲覧するツール

大島 裕樹<sup>†1</sup> 宮下 芳明<sup>†1†2</sup>

私たちは多くのアプリケーションやウィンドウを立ち上げ、切り替え、そして閉じながら PC で作業している。画面が刻一刻と変化し、次々と情報が現れては消えてゆく中を、ユーザはわずかな短期記憶を頼りに渡り歩いている。そしてついさっき見たはずの情報であっても、それを確認しようとする情報源は既に消えてしまっていることも多い。このようなとき必要となるのは、アプリケーション毎のアンドゥや、大がかりなバックアップ/復元のように操作を取り消す機能ではなく、あたかも短期記憶を拡張するかのように、過去の状態を自由に閲覧して思い出させる機能であると筆者らは考えた。そこで本稿では、簡単なマウス操作でデスクトップの任意の領域をくり抜いて、反時計回りのドラッグ操作によって時間を巻き戻し閲覧するツールを提案する。閲覧範囲は自由に変形したり半透明化したりもでき、過去のマウスカーソルも閲覧することができる。

## Rewind-ow: A Tool to Rewind Time of Specific Area and Browse

YUKI OSHIMA<sup>†1</sup> HOMEI MIYASHITA<sup>†1†2</sup>

Many people work on our PC using many applications and they open a lot of windows, pass from one to another and close them. They walk through the windows relying on short term memory while the screen changes every moment and the information appear and disappear in sequence. Sometimes they have troubles to look back the information that they just saw, because the source of information has already disappeared. What is required in such situation is not undo function for each application, or large scale backup/recovery to nullify operations. We think what is needed here is a function to remind the past state by freely browsing, as if expanding the short term memory. We propose the system to hollow out the arbitrary area on the desktop and rewind and browse time by mouse dragging counterclockwise in a circle. Browsing area can be transformed freely or made translucent and even mouse operation is browsable.

### 1. はじめに

コンピュータを使用していて、少しだけ前の状態を閲覧したいと思うことは多い。登録時に提示された番号を控え忘れてしまい、直後のログイン時に確認をしたくなかった時や、Twitterのように随時流れていくコンテンツで、少し前に見た発言をもう一度見たい時などである。しかしコンピュータのディスプレイ上では、常に表示される情報が更新され、過去に表示されていた内容は消えてしまう。たとえ直前までそこに映っていたはずの情報であっても、情報源を消してしまえばその情報を確認することは容易ではない。筆者らはこれまで、「時間の窓」を用いて過去の状態を覗き見る手法を提案してきた[1]。この手法では、ユーザは画面に時間の窓を生成し、図1の様にその窓の範囲内に過去の状態を表示できる。部分的に過去の状態を映し出すことで、その内容を閲覧しながら現在の作業を行える。

コンピュータ上で過去の状態を再現する手法は多く存在する。代表的なものとしてUndo機能が挙げられる。Undo機能は直前の操作を取り消すもので、これを繰り返すことでアプリケーション上での過去の状態を復元することができる。また、ファイルのバックアップを利用した復元機能

も過去への移動の一例である。ファイルの複製を作成しておくことで、編集などに失敗した際にその前の状態を再現することができる。MacOSに搭載されているTime Machine[2]ではファイルバックアップを用いて、まるで過去から現在へとファイルを移動させるかのような操作での状態復元を実現している。

これらの機能はユーザの操作ミスを取り消したり、編集を行う前の状態を容易に再現したりと非常に有用であるが、その一方で制約も多く手軽さに欠ける部分がある。復元機能で移動可能な過去の状態はファイルの複製を行った時点に限られ、ユーザが事前に準備した過去にしか移動はできない。Undo機能で移動ができるのはファイルを開いてからの時間のみであり、一度ファイルを閉じてしまうと過去の状態への移動は不可能になる。筆者らの提案してきた過去状態の閲覧手法では、過去の状態への干渉は行わずに閲覧のみに限定することで、大量の過去状態を高速で保持し、操作をマウスのみ限定することで、マウスに機能が追加された感覚で、現在の作業からのシームレスな使用を実現してきた。

本稿では、「時間の窓」での過去の閲覧手法に対して得られた意見から機能の改善・追加を行い、よりユーザの必要とする過去の閲覧を実現したシステムRewind-owを提案する。これまではユーザの記憶を頼りに過去を閲覧したい部分を指定し、部分的に過去の状態を表示することで過去を

<sup>†1</sup> 明治大学理工学部情報科学科  
Department of Computer Science, Meiji University

<sup>†2</sup> 独立行政法人科学技術振興機構, CREST  
JST, CREST

閲覧しながらの作業を実現してきた。しかしユーザが表示位置まで記憶に残しておく期間はそれほど長くはなく、時間が経つほどに記憶は曖昧になってしまう。そのような場合部分的に過去を閲覧する手法では、必要とする情報が表示されている場所を見落とす可能性が高くなり、部分的閲覧の利点が活かせない。そこで、時間の窓による過去の閲覧を「数分前」のようなごく最近に特化し高速化を図り、遠い過去の閲覧に対しては別の操作法によって閲覧を行う手法を提案する。

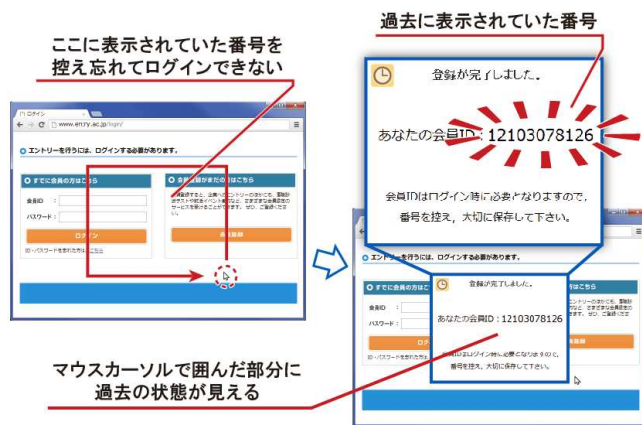


図 1 既発表システム「時間の窓」[1]を用いた過去の閲覧  
Fig. 1 Total past using "time window" [1]

## 2. 関連研究

過去の状態への移動を扱った研究として、暦本が提案する Time-Machine Computing がある。これはコンピュータ上の状態を時間ごとに保存し、それらを自由に移動する手法により、過去の状態の再現を可能としたものである[3]。過去の状態を再現するには、ファイルの操作などのイベントにスナップしての時間指定が可能であり、コンピュータ上で行われた操作を視覚化することでユーザが求める時間への移動をスムーズに行えるようになっている。Time-Machine Computing では2つの配置手法を取り入れている。ひとつは Malone らの Rooms[4]に見られる空間配置の手法であり、ファイルの位置によるグループ分けや電磁的付箋、に意味を持たせる方法である。もうひとつは Freeman らの Lifestreams[5]に見られる時間順配置の手法であり、すべてのファイルを一次元の時間軸上に置く方法である。Time-Machine Computing では、この2つを相補的なものとして扱い、コンピュータの状態そのものを時間順に管理している。本稿はこの Time-Machine Computing の考え方を取り入れ、配置などの状態を含めて、過去の状態を閲覧することを目的としたものである。

近藤らの Retrospector[6]では、PC 内での作業を記録してデスクトップ上の状態の画像を作成し、ユーザに過去の状態を閲覧させることができる。この研究では、過去の自分の履歴を簡単に振り返ることが、有効に働く場面が多いと

している。コンピュータ上では、ユーザは多くのアプリケーションを組み合わせて作業をすることが多く、ひとつのアプリケーションの履歴のみでは、ユーザにとって不十分と感じることも多い。例えば、高機能なアプリケーションを使用する際に、インターネットブラウザで使い方などを調べながら作業することは多い。Retrospector では、作業の様子を画像で閲覧するといった簡単な手法で、過去の自身の情報を取り出すことを可能にしている。本稿での提案手法は Retrospector の目的と共通する部分が多い。ユーザは自身の持つ記憶を頼りに過去の状態を閲覧し、必要な情報を確認することができる。

領域を指定して部分的に過去の状態を復元する研究は数多く存在し、Kawasaki らの Regional Undo for Spreadsheets ではスプレッドシート上の選択した領域の Undo を実現しており、必要な部分のみ状態を復元することを可能にしている[7]。同様の研究として Prakash らはテキストエディタにおいて部分的 Undo を実現しており、ユーザが指定した範囲のみ Undo することのできるシステムを提案している[8]。Thomas らは、多人数での作業を対象とした部分的 Undo を提案しており、近くアイテムをグループとしてその中で Undo する手法や、ユーザごとの作業領域内での Undo をする手法、ユーザの視界に応じて Undo する範囲を指定する手法を提案している[9]。大江らの提案する手法では、記録したマウス操作の軌跡を視覚的に表現し、ユーザが Undo したい部分の軌跡をなぞることで、なぞられた箇所までの Undo を実現している[10]。これらの研究で提案された手法は、Undo したい場所が分かっている時に非常に有効なものであり、既存の Undo 機能では直前のコマンドを取り消す操作をしていたのに対して、場所を指定することでより直感的に Undo することができる。

映像内の一部分の時間操作を表現したものとして Khronos projector があり、映像が投影されたスクリーンに触れることで、触れた部分の時間を変化させることができる[11]。時間の変化はスクリーンを強く押すほど大きくなり、ユーザの力加減で任意の時間を閲覧することもできる。本稿は Khronos projector の、部分的に時間が変化するという発想をコンピュータの画面上で実現したものであり、スクリーンを押す代わりに、マウス操作によって時間の移動を行う。

## 3. システム

本章では、これまでの提案手法におけるシステムと、Rewind-ow に新たに追加された機能について述べる。

### 3.1 ディスプレイの状態保存

本システムではディスプレイの状態を常に画面キャプチャしており、記録は一秒間隔で自動的に行う。キャプチャしたメモリ上に保持し、同時に画像ファイルとしての保存も行う。メモリ上に保持されている状態は、時間の窓の生

成によって近い過去を閲覧する際に使用する．画像読み込みの負荷を排除することで，より快適に過去の状態の閲覧が行えるようになる．メモリ上に保持できる限度を超えた場合，古い状態から順番に削除していき，ファイルとして保存された画像のみが残る．保存した画像ファイルは遠い過去を閲覧する際に使用するため，自動的に削除することはしない．

キャプチャの対象は画面に映っているものすべてであり，ウインドウの配置などを含めてすべて画像として記録される．マウスカーソルの位置に関しては，通常のスクリーンキャプチャでは記録されないため，システム側でキャプチャ時のマウスカーソルの位置を取得しシステム側で用意したマウスカーソルを追加で描画する．ユーザがマウスカーソルの存在に気づきやすいように，通常のマウスカーソルに比べて目立つものを描画する．この時，図2のようにマウスカーソルの軌跡を含めて描画することで，その瞬間にユーザがしていたことが分かりやすいようにしている．

画像ファイルを保存する際には，記録時の時間情報をファイル名とする．メモリから削除された遠い過去を閲覧する際にはファイル名をもとに時間順に並べ画像を表示する．

記録時のマウスカーソルの位置



図2 マウスカーソルの軌跡の追加表示

Fig. 2 Additional display of the locus of the mouse cursor

### 3.2 時間の窓の生成

時間の窓はマウスの操作によって生成する．右ボタンを押したままマウスカーソルで図形を描くと，描いた図形と同じ形の時間の窓が現れる．これまでの提案では，時間の窓は円形に限定されていたが，コンピュータ上ではウインドウなど四角形で表現されているものが多く，円形では過去状態を表示した時に文字が縁で欠けてしまうなど，閲覧上の問題が生じていた．また時間の窓の形が限定的になることで，ユーザが作業を行なっている場所にまで時間の窓が広がってしまうという問題もあったため，今回自由図形での生成を行えるように改善した．

時間の窓はメモリに確保してある画像から，マウスカーソルで囲われた部分以外を非表示にすることで実現している．時間の窓から見える過去の状態と，現在の状態の座標

を完全に一致させるために，画面と同じサイズの非表示ウインドウを予め生成しておき，メモリに記録されている過去の状態を表示している．本システムでは特定の色を透過色として扱っており，図3のように表示しない部分を透過色で塗りつぶすことで過去の状態の部分的な表示を行なっている．透過色は今回 RGB 値で(10,15,20)とし，コンピュータ上で使われる頻度の低い数値を指定した．

生成された時間の窓は右ボタンを離れた後も残り，時間の窓から過去を覗き見たまま現在の作業を進めることができる．時間の窓はマウスの左ボタンを押したままドラッグすることで移動でき，閲覧している場所と，現在作業を行なっている場所が重なってしまった場合にも時間の窓が邪魔にならないようにして作業することができる．生成した時間の窓は，窓の上で右クリックをすることで消すことができる．時間の窓が消えるまでは，後述する閲覧内容の変更操作を随時受け付ける．

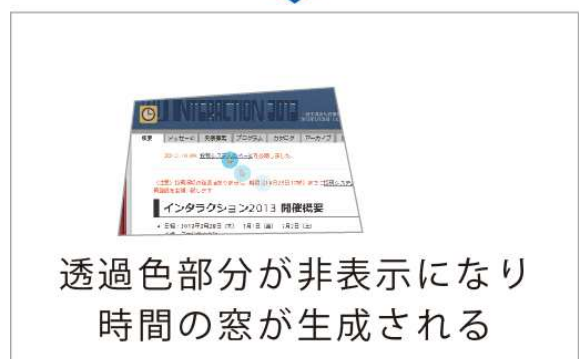


図3 時間の窓の生成手順

Fig. 3 Procedure of generating the time window



### 3.3 閲覧時間の指定

生成した時間の窓に対して、右ボタンを押したままマウスカーソルを回転させることで閲覧する時間を指定する。マウスカーソルの回転方向と表示される時間は、時計の針の移動方向に対応した操作方法をとっており、反時計回りに回転させるほど古い状態が閲覧でき、時計回りに回転させるほど新しい状態が閲覧できる。

マウスカーソルの回転方向は3点の座標から計算して求める。マウスカーソルの位置は常にシステム上で取得しており、現在の位置を含めた過去の3点を用いて計算を行う。計算して求めた値が正の数ならば反時計回り、負の数ならば時計回りと判断し、時間の窓から見える状態へと反映させる。

時間の指定は右ボタンを押した場所からの角度によって指定を行う。この時マウスカーソルを回転させる位置によって、時間の窓から見える状態を変化させるのに必要な回転角度が変化している。変化の割合は3段階で変化し、近距離では60度で1段階、中距離では30度で1段階、遠距離では10度で1段階変化している。大きく時間を変化させたい場合は遠距離で回転させ、細かく時間を指定したい時は近距離で回転させれば良い。回転位置の基準は時間の窓の中心とし、時間の窓の大きさや位置に左右されずに時間の指定を行える。基準となる点には図4のようにダイヤルを表示する。このダイヤルは時間の移動を行なっている時のみ表示され、マウスカーソルを回転させる場所の目安になる。また、ダイヤルの中央には現在表示している状態の時刻を表示し、時間の窓から見える状態に変化が見えづらい時にもユーザに時間の移動が伝わるようになっている。

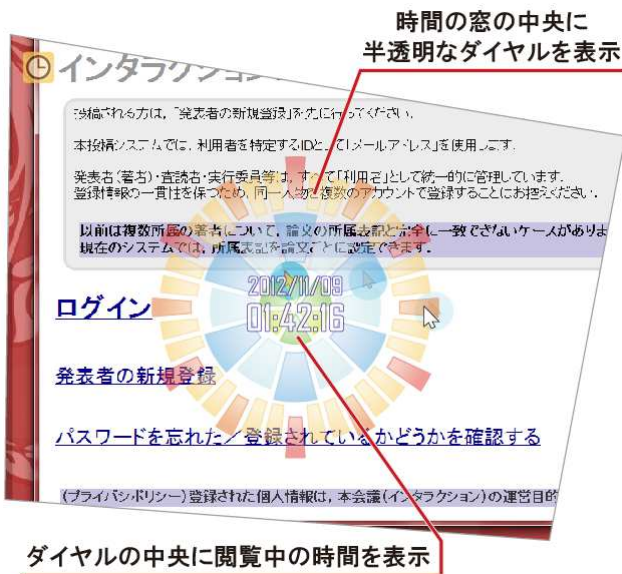


図4 基準を示すダイヤル

Fig. 4 Dial to indicate the criteria

### 3.4 時間の窓の自由変形

時間の窓を生成した後に、閲覧範囲を変更したくなることも考えられる。時間の窓の操作は、左ボタンに表示位置の変更、右ボタンに表示する時間の変更が対応しているため、自由変形を行うときにはモードの変更を行う。時間の窓を生成すると、図5のように隅にモード切り替えボタンが表示される。自由変形モード時は、右ボタンを押しながらマウスカーソルで図形を描くことで、現在表示している時間を維持したまま時間の窓の形を変更することができる。また、左ボタンを押したままドラッグすることで時間の窓を動かしながら、覗き見る場所を変更することができる。

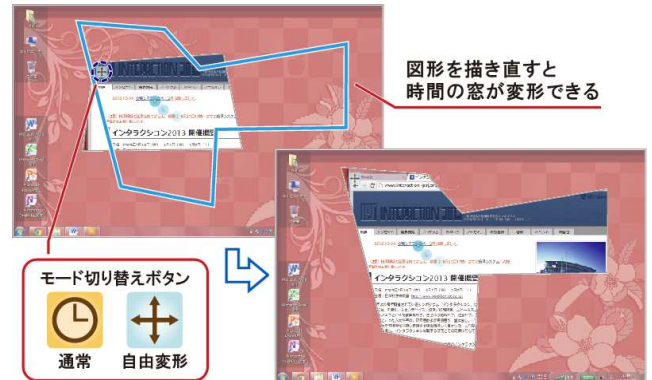


図5 自由変形モード時の時間の窓の移動

Fig. 5 Moving time window during the Free Transform mode

### 3.5 時間の窓の半透過機能

時間の窓から覗ける過去の状態は、図6のように表示の濃さを調節でき、半透明な状態で過去の状態を重畳表示することができる。この機能を利用することで、過去の状態と現在の状態を重ねあわせながら編集した場所を探したり、絵などを描いている際に編集する前と後での違いを比較したりといった用途にも使用できる。半透明の時間の窓は、ウィンドウスタイルに透過ウィンドウスタイルを指定し、透過率をリアルタイムで変更することによって実現している。透過率の変更はマウスのホイールで指定する。

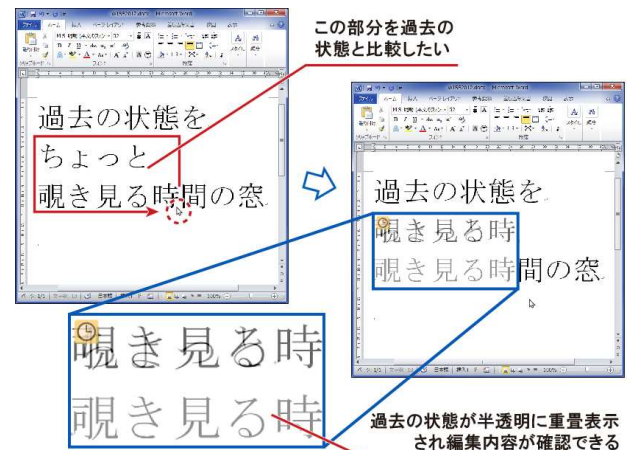


図6 半透過による過去の重畳表示

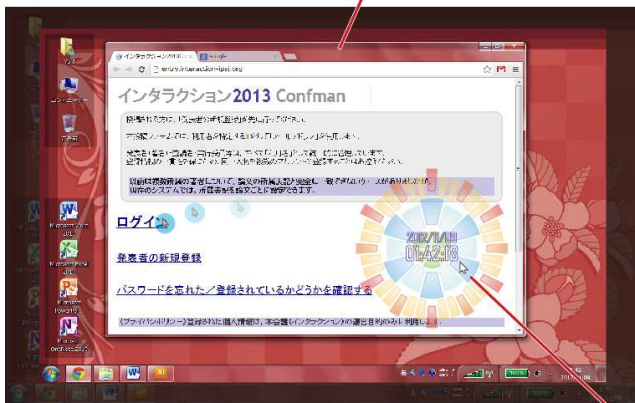
Fig. 6 Superimposed by a semi-transparent view of the past

### 3.6 時間の窓を用いない遠い過去の閲覧

メモリから削除された状態を閲覧する際には、画面全体で過去の状態を閲覧する。マウスで右ダブルクリックを行うと、図7のような画面全体での過去状態閲覧状態へと移行し、時間の窓での時間指定と同じ操作方法で閲覧する時間を指定することができる。この状態では過去の状態を閲覧しながらの作業を行うことができないため、ユーザはマウスの左ダブルクリックをすることでその状態の画像ファイルをデスクトップに生成することができる。

この機能は過去に生成したファイルの読み込みが必要となるため、時間の窓を用いての閲覧に比べて時間を必要とする。あくまでも、提案手法での時間移動操作を生かしたうえで、遠い過去についての閲覧をサポートするための機能である。

一回り小さく過去の状態が表示される



時間の窓を用いた時と同じ操作法で時間を移動

図7 画面全体での閲覧方法

Fig. 7 Method for viewing the entire screen

## 4. 提案手法の解決する問題例

本章では Rewind-ow を用いて解決を図る想定問題と、その解決法について述べる。

### 4.1 番号を控え忘れた場合

ユーザ登録時に固有の番号が与えられ、ログイン時に入力を求められることは多い。特に登録時にメールアドレスなど連絡手段を必要としない場合、確認するためには問い合わせなどの手段を用いるしか無く、非常に手間と時間がかかってしまう。

番号を控え忘れた直後に気がつけば、番号が表示されていた場所に時間の窓を生成し、番号が確認できたらテキストエディタなどでメモをとることで、問題を解決することができる。番号を控え忘れたことに気がつかず、ある程度の時間が経ってしまっている場合には、画面全体での閲覧方法を用いて番号が表示されている状態を探し出せば良い。番号をテキストエディタなどでメモしたい場合は、過去の状態画像をデスクトップに一度保存し画像ファイルとして

閲覧しながらメモを取れば良い。

### 4.2 Twitter で直前に見た発言をもう一度見たい場合

Twitter は常に情報が流れていくコンテンツである。新たに発言があった場合、手動または自動で更新を行うことで新しい発言を読むことができる。常に大量の発言が投稿されるコンテンツであるため、お気に入りへの登録などを行わない限り、曖昧な記憶から同じ発言を再び見つけることは困難である。またクライアントによっては、更新と同時に発言が移動してしまうものもあり、不意に自分の見たい発言を見失ってしまうこともある。

提案手法で Twitter 閲覧時の状態を見ることができれば、自分の探している発言を確認することができる。確認した後に発言の一部を検索すれば、Twitter 上で発言を見つけ出すことも容易であり、お気に入りに登録するなどして以後見つけやすくすることもできる。

## 5. おわりに

本稿ではこれまで筆者らが提案してきた、過去の状態を部分的にかつ手軽に閲覧することのできる手法を改善し、既存機能の改善と新たな機能の追加を行ってきた。結果としてユーザの求める時間移動をより快適に、より手軽に実現できる手法を提案することができたと感じている。

今後は既存の手法との比較を行い、どのような場面で本稿での提案手法が有効であるかを考察していく必要がある。現状では時間の窓から見える文字列やファイルを現在へと反映させることは実現していないが、将来的には文字認識やファイル管理などを取り入れ、より多くの対象について時間の移動を実現したシステムを構築して行きたいと考えている。ファイルを誤って消去してしまった場合、現在のシステムでも過去の状態を閲覧しながらユーザが状態を再現することは不可能ではない。しかし、完全に再現することができるのは文章やプログラムなどの文字によって構成されているものに限られ、画像ファイルに関しては完全に再現することは難しい。なによりコンピュータ上において手作業で状態を復元する行為は、データの移動や複製が手軽に行える環境を全く活かしていないと感じられる。多機能化することで、本稿で実現した手軽さが損なわれる可能性はあるものの、MacOS の Time Machine のようにユーザインタフェースを工夫することで、システムの複雑さを感じさせない操作を実現したいと考えている。

本稿の提案手法に対する用法はあくまで一例であり、この手法によってユーザが得られるものに関しては、すべてを把握できているとは言い切れない。提案手法に対してどのような活用法が存在するかについては、多くのユーザに使用してもらい、今後も議論を重ねていきたい。

また本稿の提案手法は、スマートフォンのように画面サイズが限られており、マルチウィンドウの実現していない環境でこそ効力を発揮すると考えている。大型のディスプ

レイを持つコンピュータでは、複数のウインドウを並べることも容易であり、大事な内容であればウインドウを閉じること無く作業を進めることもできる。対してスマートフォンのような端末では、複数のウインドウを並べての作業は困難であり、常に作業中のウインドウを最前面においての操作を余儀なくされている。このような環境であれば、部分的に過去の状態を閲覧する行為は、擬似的にマルチウインドウを実現しているとも言え、複数のアプリケーションを行き来しながら作業をした場合に比べて、大きく効率を上げることができるのではないかと考える。

## 参考文献

- 1) 大島 裕樹, 宮下芳明. 過去の状態をちょっと覗き見る時間の窓, WISS2012 論文集, 2012.
- 2) TimeMachine, <http://support.apple.com/kb/HT1427>.
- 3) 唇本 純一. Time-Machine Computing 時間指向ユーザインタフェースの提案, WISS1999 論文集, pp.55-64, 1999.
- 4) Thomas W. Malone. How do people organize their desks? implications for the design of office information systems. ACM Trans. On Office Systems, Vol. 1, No. 1, pp. 99-112, 1983.
- 5) E. Freeman and D. Gelernter. Lifestreams: A storage model for personal data. ACM SIGMOD Bulletin, March 1996.
- 6) 近藤 秀樹, 三宅 芳雄. 計算機上での活動履歴を利用する記憶の拡張システムの評価, WISS2005, 2005.
- 7) Yoshinori Kawasaki, Takeo Igarashi. Regional Undo for Spreadsheets, UIST2004 demo, 2004.
- 8) Atul Prakash and Michael J. Knister. A framework for undoing actions in collaborative systems. ACM Transactions on Computer-Human Interaction, 1(4), pp.295-330, 1994.
- 9) Thomas Berlage: A selective undo mechanism for graphical user interfaces based on command objects. ACM Transactions on Computer-Human Interaction, 1(3), pp.269-294, 1994.
- 10) 大江龍人, 志築文太郎, 田中二郎: 軌跡に基づいた undo/redo インタフェース, 情報処理学会研究報告 HCI, ヒューマンコンピュータインタラクション研究会報告 HCI-149, No.7, pp.1-8, 2012.
- 11) Cassinelli, A, Ishikawa, M. Chronos projector, ACM SIGGRAPH 2005 Emerging technologies, SIGGRAPH'05, New York, NY, USA, ACM, 2005.