Vol.2012-HCI-148 No.3
2012/6/1

トランプ型ファイルブラウジング手法の提案

鈴木悠司[†]　井川洋平[†]　宮下芳明[†,††]

従来のエクスプローラでのファイルの閲覧・探索は，階層構造によるフォルダ表現と，階層ごとに分割されたファイル表示によって行われていた．これら二つはファイル探索に大きな手間を取らせる要因だと考える．そこで，本稿ではトランプ型ファイルブラウジング手法を提案する．本手法により，フォルダの構造を俯瞰的に認識しながら，探し出したいファイルを見つけることが可能となり，ファイル探索行為の手間を軽減し，効率的にファイルを見つけ出すことができる．

File Browsing in Playing-Card-Style

YUJI SUZUKI[†]　YOHEI IKAWA[†]　HOMEI MIYASHITA[†,††]

In Windows-explorer-like file browsing, folders have hierarchic structure and they are displayed in discrete windows, and we think those worsen the usability of file browsing. In this paper we propose a new way to browse files that is fuss-free and efficient; a style like playing card. Using this methodology, the user can find any files in any folder, and also can get the picture of the whole file structure.

1. はじめに

グラフィカルユーザインタフェース（GUI）におけるファイルの閲覧・探索は通常，階層構造によるフォルダ表現と階層ごとに分割されたウィンドウでのファイル表示によって行われる．本来，ファイルを探す行為は何かの作業中に発生する副次的なものであり，行為そのものが簡潔に行え，かつ短時間で目的のファイルを見つけられるべきだと筆者らは考える．

フォルダを階層構造内の一階層と考えた際，現在のエクスプローラによるウィンドウ表現は，その時見ているフォルダ（フォーカスしている階層）の内容を明確に表現しているが，エクスプローラで開かれていない他のフォルダの中身はわからない．つまり，フォーカスされている階層は，フォーカスされていない階層から独立して表示されている．そのため，ファイル探索時のように今見ている階層と別の階層の情報を知りたい場合では，フォーカスしている部分を切り替える必要がある．

階層を移動する場合において，二つの操作方法がある．一つはウィンドウに表示されている情報を切り替える操作，もう一つは複数ウィンドウを用いて，複数の階層を同時にフォーカスする操作である．ウィンドウ表示の切り替えでは，フォーカスされている階層内にあるフォルダを開くか，エクスプローラ左上にある「戻る」ボタンなどにより前のフォーカスに戻す操作をしなくてはならない．本稿ではフォルダを開く操作を「階層をもぐる」行為，戻る操作を「階層を上がる」行為と割り当てる．ファイルを探す際には，これらの「もぐる」・「上がる」といった操作回数がフォルダ構造と GUI の関係上多くならざるを得ない．沢山のフォルダが入れ子状に置かれ，階層構造が複雑化するにつれ，これは顕著になる．複数の階層を同時にフォーカスする操作でも，該当するウィンドウを探索するまでは，階層を「もぐる」・「上がる」行為を繰り返さなければならない．さらに，ウィンドウ同士の位置を動かす操作，アクティブウィンドウを切り替える操作の回数も増えてくる．階層構造が複雑化するに伴い，新規ウィンドウの生成やウィンドウの移動など，ファイル探索とは直接関係のない別の操作が必要になり，結果的に探索時の手間が増える．

また，一階層のみのフォーカスでは全体俯瞰が難しいという問題がある．全体の構造がつかめないために，なかなか目当てのファイルが見つからず，探索に長い時間を費やすことも少なくない．従来の GUI では，局所的なフォーカスによって全体俯瞰が難しくなるだけでなく，階層の移動やそれに伴った操作の手間がかかってしまう．

本稿ではファイル探索行為の手間を軽減し，効率的にファイルを見つけ出すためのフォルダ構造及びファイルブラウジング手法として，トランプ型ファイルブラウジング手法を提案する．この手法は，全ファイルを俯瞰的に認識しながら局所的な拡大を行い，効率的にファイルを見つけ出すことを目的としている．

以降では，提案手法のデザインコンセプト及びインタフェースデザイン，そして実際のプロトタイプ及び操作方法について述べる．

2. 提案手法

提案するトランプ型ファイルブラウジング手法について，デザインコンセプトを述べる．本手法は focus+context 手法に基づいて，フォルダ構造の表現とファイルブラウジング手法の二つから構成される．

[†] 明治大学理工学部情報科学科
Department of Computer Science, Meiji University
[††] 独立行政法人科学技術振興機構, CREST
JST, CREST

2.1 フォルダ構造の表現

提案手法では階層構造全体を常に表示することで，あらかじめ全ファイルを表示できるようにしている．一階層の表現として，トランプの束が一直線に展開されているかのようにフォルダ内部の情報が常に見える状態になっている．このファイルの束を，以降フォルダと呼ぶ．ファイル束の長さで，フォルダ内にどれくらいの数のファイルが含まれているかが一目でわかるようになっている．

フォルダ内に更にフォルダが存在する場合，上位階層のフォルダに直交する形で下位階層のフォルダが展開される（図 1）．これにより，階層ごとにウィンドウが分割される問題点を解決し，同時に，全体のフォルダ構造を俯瞰的に認識することが可能になる．図 1 はフォルダ内にフォルダが存在する 2 次階層での表現だが，提案システムは従来の GUI 同様，フォルダ内にフォルダがあり，更にその中にフォルダが存在するといった n 次の階層構造も表現可能である．

提案システムのデザインの都合上，後述するフォルダ内の拡大時において，フォルダ同士が衝突する可能性を考慮する必要がある．この問題に対して，衝突する可能性のあるフォルダ内でファイル同士の間隔を調節することにより，フォルダ同士の衝突を回避するようにした（図 2）．これはファイルの拡大表示時にはその拡大状態に応じて長さを調節する機能である．

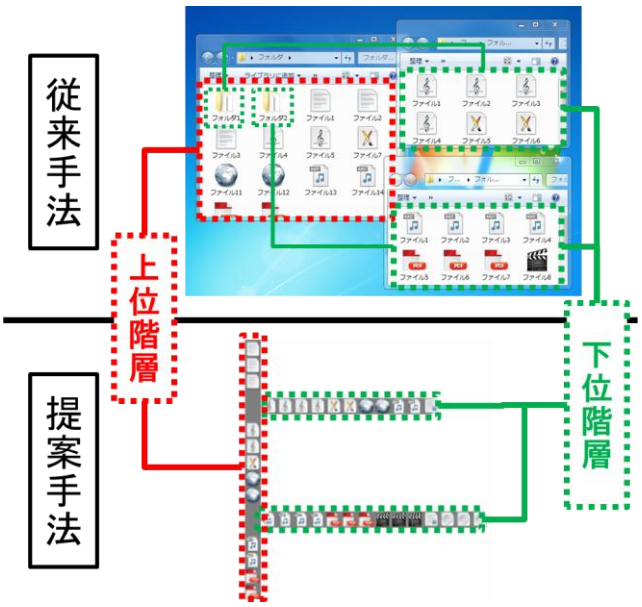

図 1　フォルダ構造
Figure 1　Folder Structures in Conventional Method and Proposed Method.

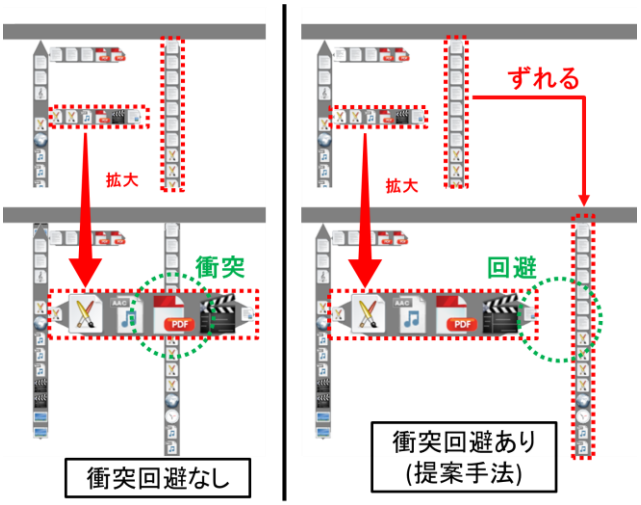

図 2　フォルダ同士の衝突回避
Figure 2　Avoiding Collision between Folders.

2.2 ファイルブラウジング

従来の GUI における一階層を表示する方法について，操作の手間を増やしてしまう問題がある．例えば，ウィンドウ表示サイズを超える数のフォルダやファイルがある場合，ウィンドウはスクロールバーを表示して，現在見えているフォルダやファイル以外にも，このフォルダ内には存在することを表現していた．これはフォルダやファイルのアイコンが一定の大きさを維持しており，それらの数や，ウィンドウサイズによって大きさを個別に変えられなかったからである．この場合，スクロールしなければ同一階層内に探していたファイルがあるにも関わらず，ユーザはそのファイルを見落としてしまう可能性があった．

これに対し提案手法では図 1 に示したように，フォルダはアイコンではなくファイルの束として表現されており，常にフォルダ内部が展開され，全フォルダ・ファイルが確認できる状態になっている．ファイル表示に関してはあらかじめ全体のフォルダ構造を維持できる程度まで小さくアイコンで表示し，特に確認したい部分がある場合は，その領域を拡大することで，選択領域内のファイルアイコンの大きさを変更する．図 3 に概略図を示す．全ファイルの一覧の表示をしている状態で局所的な拡大を行うことにより，効率的にファイル探索が行える．本システムで表示できるファイル数は，PC での閲覧を考え 10,000 個程度を想定している．これは 17 インチ程度の PC ディスプレイを想定した個数であり，対応するハードウェアが変わった場合には，そのディスプレイサイズに応じたファイル数に変わるものとする．

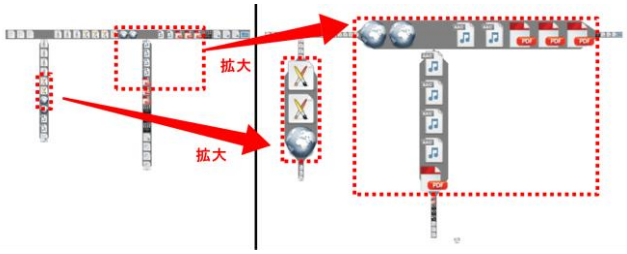

図 3　ファイルのブラウジング
Figure 3　Browsing Files.

3. プロトタイプシステム

提案手法のプロトタイプシステムを JavaScript で実装した．図 4 にシステムの初期画面を示す．基本的な操作の手順は次のようになる．

① 確認したいファイルの拡大
② 拡大領域の移動

以下で各手法について説明する．

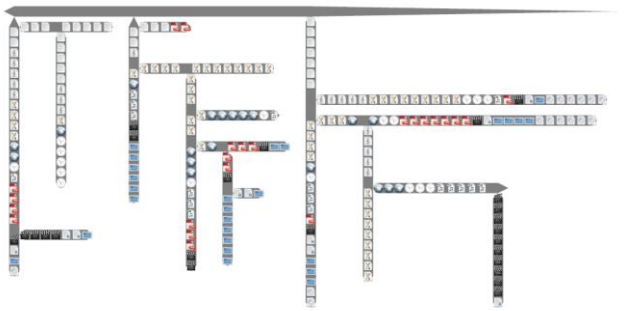

図 4　システム初期画面
Figure 4　Initial Screen of Proposed System.

3.1 矩形領域指定型拡大手法

矩形領域指定型拡大手法では，確認したいファイルの拡大表示と拡大解除を行う．この操作では，マウスの水平移動で拡大する領域を決定し，垂直移動で拡大倍率を決定する．マウスの右ボタン押下地点で範囲指定の中心を決定し，そこから水平移動することで半透明の矩形を描画する．これにより拡大操作の対象となる領域を指定する．右ボタンが押下された地点を基準点とし，そこからの水平方向の移動距離で範囲の一辺の長さを指定する．基準点から遠ざかるほど領域は拡大される．この操作により指定された範囲内に含まれるファイルが拡大表示操作の対象として選択される．選択領域の形状として矩形以外にも円形などが考えられるが，フォルダの配置の都合上，矩形領域による指定を採用した．

図 5 に指定領域の拡大と拡大解除方法の例を示す．右ボタンを押下し，その後水平方向に動かすことで矩形を描画し，拡大表示を行う範囲を選択する．この状態から，垂直方向にマウスをドラッグすることで，範囲内のファイルアイコンの拡大倍率を変更することができる．垂直方向のマウスの移動量に応じて拡大倍率が上昇する．

また，何もない領域で右クリックをすることで，全ファイルの拡大を解除する．これにより，目的のファイルを見つけ次第, 不要になった拡大領域を元に戻すことができる．

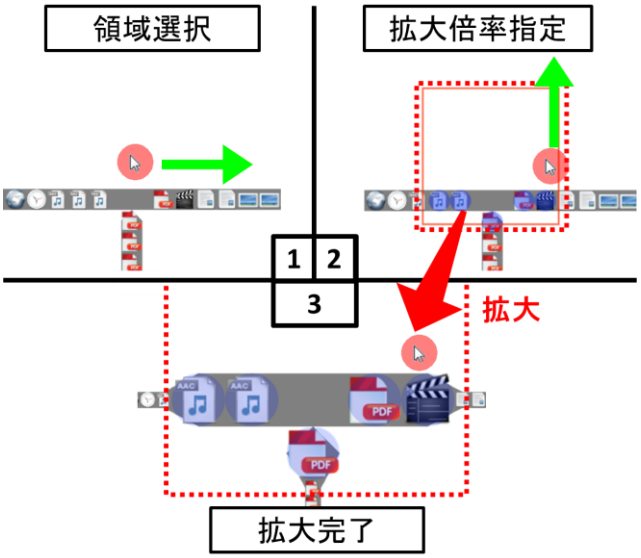

図 5　矩形領域指定型拡大手法
Figure 5　Zooming by Range Specification.

3.2 フォルダ内領域選択型拡大手法

フォルダ内領域選択型拡大手法の操作はマウス左ボタンにより行う．マウス左ボタンを押下したままマウス軌跡がフォルダを横切ることにより操作対象となるフォルダを選択する．フォルダを横切るとは，図 6 のような操作を指している．複数のフォルダを横切る場合，横切ったフォルダ全てが以降の操作の対象となる．

この手法ではこの操作の後，更に U ターンをして，再度同一フォルダを横切ってからフォルダと並行にマウスを移動することで，フォルダ内のファイル指定と拡大操作を行うことができる．

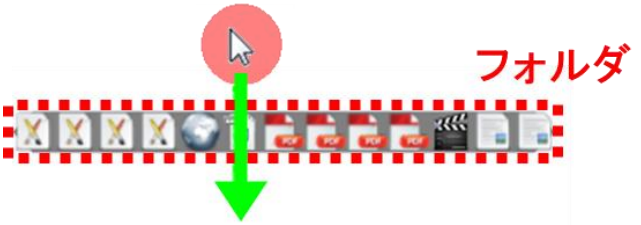

図 6　フォルダを横切る動作
Figure 6　Gesture that Cross Files.

左ボタンを押下した状態でフォルダを横切り，操作対象となるフォルダを選択する．この後，U ターンして再度フ

ォルダを横切ることでフォルダとマウス軌跡との間で二つの交点が得られる．フォルダ内のこの二交点で区切られた区間内にあるファイルが，拡大表示の対象範囲になる．対象範囲が定まった状態でそのままフォルダと平行にマウスをドラッグすることで，ファイルアイコンを拡大表示する（図 7）．また，複数のフォルダを横切ることで同時に拡大ができる（図 8）．

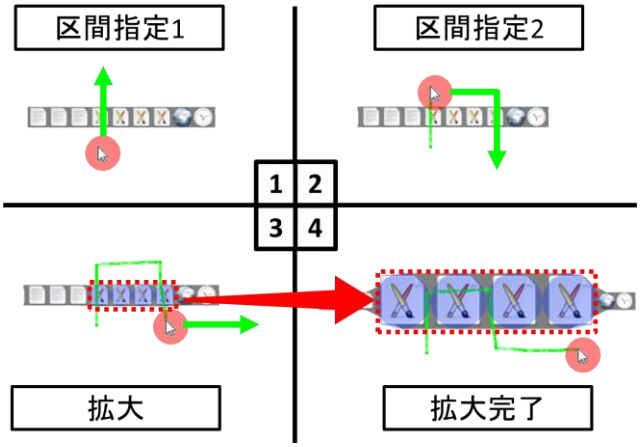

図 7　フォルダ選択型拡大手法
Figure 7　Zooming by Selecting the Folder.

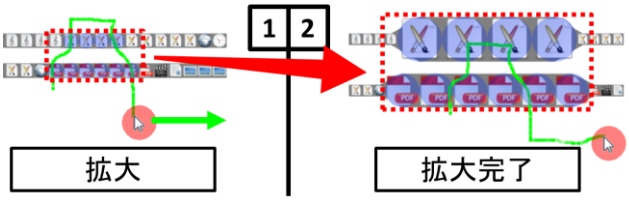

図 8　複数の領域の同時拡大
Figure 8　Zooming by Selecting Folders.

3.3 拡大領域の移動操作

拡大領域の移動操作は，マウス左ボタンにより行う．3.2 と同様に，マウス左ボタンを押下したままマウス軌跡が拡大領域を横切ることにより操作対象となる拡大領域を選択する．操作対象の拡大領域が選択された状態でフォルダと平行にドラッグすることで，拡大領域が移動する．水平方向に展開されたフォルダに対する操作を説明する．図 9 で示すように，拡大表示している領域がフォルダと平行に移動する．領域はマウスの左右移動に対応した方向に移動する．二つ以上の拡大領域を選択する場合でも，その全てが移動する．垂直方向に展開されるフォルダの場合は，この操作を右回りに 90 度回転させた操作となる．

また左ボタンを押下し拡大領域を横切る操作を行わず拡大領域を選択しない状態でも，何もない場所で左ボタンを押下し，そのまま左右に移動することで，全ての拡大領域が移動する（図 10）．

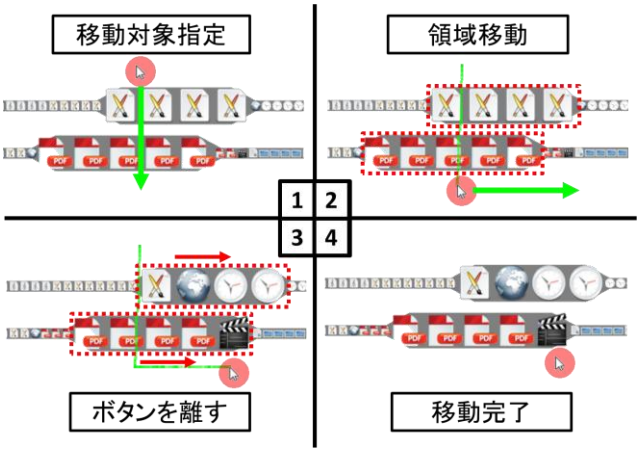

図 9　フォルダ選択型拡大領域の移動手法
Figure 9　Moving Specified Range in the Selected Folder.

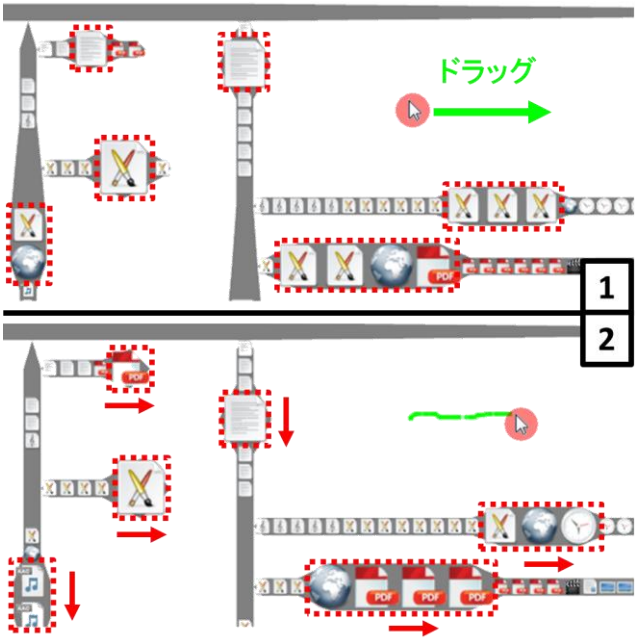

図 10　全拡大領域の移動
Figure 10　Moving the whole Range for Zooming.

3.4 拡大領域の拡散

3.3 で述べた拡大領域の移動によって，対象領域に下位フォルダが含まれている場合，その下位フォルダにも拡大が適応される（図 11）．この手法により，フォルダ内の全ファイルを効率的にくまなく探索することができる．

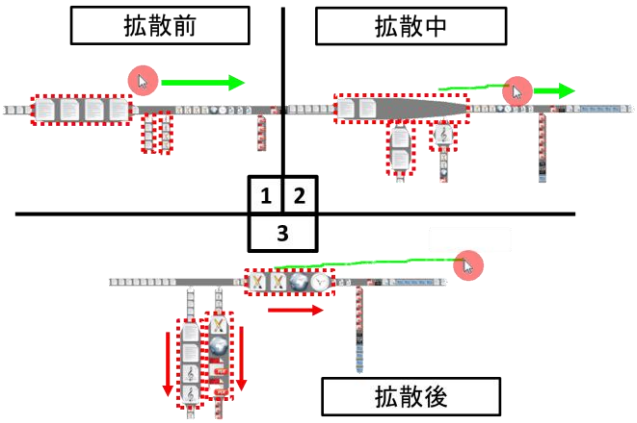

図 11　ファイル拡大の拡散

Figure 11　Spreading the Range for Zooming.

4. 関連研究

アイコンの空間的な位置を利用して自動グルーピングを行うファイル管理機構として，Watanabe らの Bubble Clusters[1]がある．このシステムでは，階層に収まりきらないグループ構造をサポートすることで，より柔軟な構造を扱うことができる．提案手法では従来の GUI 同様階層構造をそのまま表現できるようにしたが，Bubble Clusters では一次階層に限定し，下位フォルダ全体をひとまとまりのグループとみなして操作の効率化を行っている．

提案手法においてファイルブラウジングの際にズーミングユーザインタフェース（ZUI）を採用した．これと同様に ZUI を用いた動的な検索システムとして，豊田らの HishiMochi[2]がある．HishiMochi では，矩形の入れ子構造を部分的に拡大する非線形ズームという手法をとっている．これにより，全体の構造を容易に認識しながら局所的な拡大で必要な情報を得やすくしている．

伊藤らのデータ宝石箱[3]は，階層型データを二次元の入れ子構造で表現することで, 大規模なデータを一望できる．

本稿と同様に，一列に並べた項目を縦横に直交させるインタフェースとして，Sony Computer Entertainment Inc. の PlayStation®3 や PSP® における項目選択インタフェースである XMB™（クロスメディアバー）[4]がある．横軸にカテゴリが並び，その中の一項目が縦軸に展開表示されるため，上下左右のキーによってフォーカスを移動するコントローラでの正確な操作に適したインタフェースと言える．

本手法では，フォルダを一次元的な情報として表現している．これは，フォルダがスクロールバーとしての役割を併せ持っていると考えることができる．スクロール操作に関しては，柏木らの FineSlider[5]や，Ramos らの Zliding[6]がある．FineSlider は，通常のスライダと同様にスケールとノブから構成されており，ノブのひっぱり度合いに応じてスライダの移動スピードが変化するという手法を用いている．これにより，限られたスペースで広範囲のレンジを持つパラメータの値設定を，正確かつ効率的に行うことができる．Zliding では，圧力ペンによる圧力情報を利用して移動スピードの調整を行なっている．本手法では，フォルダのファイル数が増えると，通常のスクロールバー同様移動や微調整が困難となる．FineSlider や Zliding の手法を用いることで，操作自体を複雑にすることなく，スクロール操作を行いやすくできると考えられる．

また，Igarashi らの研究として，スクロール速度に合わせて自動的にズームレベルを調整するという手法[7]を用いることも考えられる．これは，高速でスクロールしているときには縮小表示，低速でスクロールしているときには拡大表示になるように調整するものであり，スクロール速度と拡大率を結びつけている点で，本稿におけるインタフェースと相性がよい．

Apitz らの CrossY[8]では，横切ると言うインタフェースを利用している．この方法により，選択操作と他の操作の複合操作をワンストロークで行うことができる. 本稿では，フォルダをスクロールバーに見立てて，これを元にフォルダ選択を実装した．もう一つの利点として，細長い対象を選択しやすいという点がある．縮小されたフォルダは非常に細長くなり，フォルダ上をクリックするといった操作が行いづらい．しかし，横切るという行為ならば，細長いものであっても容易に選択可能であり，矩形選択と異なり横切る回数といった情報を利用できる．実際，一回横切ることで拡大領域の移動，二回横切ることで拡大といった使い分けが可能となっている．

一次元に並べられたコンテンツリストの閲覧を行う手法として，山中らの研究がある[9]．コンテンツの並びに直交する方向をなぞることで，ファイルを開くといった操作を行うことなく，その場で PDF ファイルや Web ページ上のリンクなど，個々の項目内容の閲覧が可能である．本手法におけるフォルダは一次元に並べられており，項目（フォルダの中身）をファイルに限定したコンテンツリストとみなすことができる．

Schillergames. の Real Desktop© では， 3D 空間に拡張したデスクトップ上にアイコンを配置することで，自由度が高く実世界に近い操作が可能となっている[10]．本稿では，効率的な操作を行うためにあえて自由度を下げている．例えば，本手法ではフォルダを自動配置している．これにより配置の自由度は下がるが，拡大領域の移動中は配置の崩れなどを気にすることなく現在の操作に集中できる．

5. まとめと今後の展望

本稿ではファイル探索における効率化を目的とした，トランプ型ファイルブラウジング手法を提案した．本手法により，全ファイルを俯瞰的に認識しながら，局所的な拡大を行い，効率的に探し出したいファイルを見つけ出すことが可能になる．

今後は，4 章に挙げた既存のブラウジング手法との比較，評価実験を行う予定である．また，拡大手法及び拡大領域の移動手法についてのマウス操作の割り当てに関して妥当性を検討していく．

並びに，本研究ではファイル及びフォルダ管理全般における効率化を目指しており，今回のブラウジング手法のみならず，ファイルの移動，コピー，ペーストといった管理行為全般に対しても考慮していく．課題としては，拡大領域の移動に伴う画面のぶれをなくすことが挙げられる．アニメーションによる補間を用いることで，拡大時の視線移動をある程度抑えることは可能だが，高速な移動を行うと画面がぶれ，見づらくなる場合があった．

参考文献

1) Nayuko Watanabe, Motoi Washida, Takeo Igarashi, Bubble clusters: an interface for manipulating spatial aggregation of graphical objects, UIST'07 Proceedings of the 20th annual ACM symposium on User interface software and technology, pp.173-182 (2007).
2) 豊田 正史, 増井 俊之, 柴山 悦哉, HishiMochi: 非線形ズーミングを用いた動的検索システム, 第 6 回インタラクティブシステムとソフトウェアに関するワークショップ論文集(WISS'98), pp.143-152 (1998).
3) 伊藤 貴之, 梶永 泰正, 池端 裕子, データ宝石箱：大規模階層型データのグラフィックスショーケース, 情報処理学会グラフィクスと CAD 研究報告 2001-CG-104, Vol.2001, No.89, pp.65-70 (2001).
4) Sony Computer Entertainment Inc., XMB, http://manuals.playstation.net/document/jp/ps3/current/basicoperations/index.html (2012-05-07).
5) 柏木 宏一, ボーデン ジョージ, 増井 俊之, 高精細スライダ "FineSlider", 第 10 回ヒューマン・インタフェース・シンポジウム論文集, pp.297-300 (1994).
6) Gonzalo Ramos, Ravin Balakrishnan, Zliding: Fluid Zooming and Sliding for High Precision Parameter Manipulation, UIST '05 Proceedings of the 18th annual ACM symposium on User interface software and technology, pp.143-152 (2005).
7) Takeo Igarashi, Ken Hinckley, Speed-dependent automatic zooming for browsing large documents, UIST'00 Proceedings of the 13th annual ACM symposium on User interface software and technology, pp.139-148 (2000).
8) Georg Apitz , François Guimbretière, CrossY A Crossing-Based Drawing Application, UIST '04 Proceedings of the 17th annual ACM symposium on User interface software and technology, pp.3-12 (2004).
9) 山中 祥太, 宮下 芳明, コンテンツリストにおけるインタラクション手法の提案, 情報処理学会研究報告 2012-HCI-147, Vol.2012, No.11, pp.1-8 (2012).
10) Schillergames., Real Desktop, http://www.real-desktop.de/ (2012-05-07).