

PDF ビューアにおける新しいスクロール方式の提案

日大生産工 ○中尾 宗司

日大生産工 中村 喜宏

1 まえがき

現在パソコンで使われているPDFビューアは、ページが縦に並べられている縦スクロール方式がメインとなっている。しかし、従来の縦スクロール方式だと、パソコンの画面が横長であるという特徴を活かしきれていない。

横長の画面上でページを縦に並べた場合、図1の赤矢印の部分のように、画面に無駄な部分が出てしまう。見開き表示もできるが、同時に表示できるページ数は2ページが限界で、そのとき表示されるページの組み合わせも『1,2』『3,4』…のように、固定化されてしまう(図2)。このため、ページが見渡しづらく、同時に見たいページの組み合わせの切り替えも瞬時に行えない。実際、紙媒体の本の場合、少しページを捲ることで同時に4ページ以内なら参照、比較が可能[1]なため、この利点をつぶしてしまうことは問題である。また、見開き表示にした場合、文字の大きさに合わせてズームすることがある。このとき、2つのページは画面



図1 従来のPDFビューアの場合

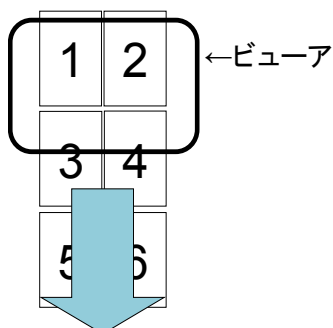


図2 見開き表示

に収まりきらずに垂直と水平、2つのスクロールバーを使う必要があり、例えば、斜め下のページに移動するときに手間がかかる。Adobe Readerにあるような手のひらツールやマウスホイールでのオートスクロールを使うという手もあるが、前者はドラッグ操作に手間がかかり、文書選択モードとの併用もできなくなってしまい、後者は速度制御が難しいといった問題が生じる。

近年では、画面の比率が4:3から16:9に変わってさらに横長になり、デスクトップ型のパソコンに至っては大画面のスクリーンを使うことが多い。このため、上記の問題がより顕著になっている。

本研究では、これらの問題を解決するためPDFビューアのスクロールの操作性に着目し、スクロール操作の多様性を増やすことを目標に、新たなスクロール方式を提案する。

2 提案方式

2-1 横スクロール

本研究では、まえがきで述べたような問題を解決するために、PDFのページを横に並べていく横スクロール方式[2]をとる。



図3 提案方式

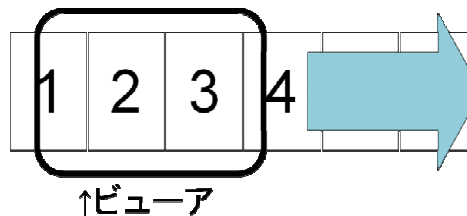


図4 横スクロールの場合

Proposal of a new scroll method for PDF viewer

Shuji NAKAO, Yoshihiro NAKAMURA

横スクロールにした場合、図3のように画面に無駄がなくなり、図1と比較しても横に長いPC画面を活かすことができていることがわかる（図3では、境をわかりやすくするために、ページを赤で囲んでいる）。

また、従来の見開き表示であった、

- ・3ページ以上同時に表示できない
 - ・『1,2』『3,4』…のように、奇数偶数の順で一度に見られるページが固定化される
- といった問題も解決される。図4が、ページを横に並べた場合なのだが、
- ・画面内に収まる限り何ページでも並べられる
 - ・横にスクロールするだけで『1,2』『2,3』『3,4』…と切り替わってくれるので、奇数偶数、偶数奇数という順になり、ページが固定化されない

といったふうに、見開き表示で述べた2点の問題が改善され、ページが見渡しやすくなっている。

2-2 クリック移動

横スクロールにすることで、縦スクロールであった問題がある程度解消されたが、ページが画面に収まりきらない場合の問題が解決されていない。まえがきでも述べたように、従来のビューアでページが画面に収まりきらないとき、

- ・垂直と水平、2つのスクロールバーを操作
- ・手のひらツール（）、もしくはそれに代替するもののドラッグ操作
- ・ホイールボタンクリックによるオートスクロール操作（）

のどれかを行うことになるが、どれも操作に手間がかかってしまう。手のひらツールは、タブレット端末でのスワイプ操作のように、ページを掴んで移動する感覚で細かい移動ができる。しかし、これをマウスでやるとなると、指でスワイプするように簡単にはいかず、押しながら移動、押しながら移動…の繰り返しになってしまうので手に負担がかかる。また、手のひらツールの最大の欠点としては、文書選択モードとの併用ができないことがあげられる。これは、手のひらツールも文書選択もドラッグ操作という点でかぶってしまっているからである。Adobe Readerの場合だと、図5にあげるように、右クリックからのポップアップメニューでいちいち手のひらツールか選択ツールかを切り替えなければいけない。この選択ツールで、マウスカーソルが文書選択用のものになる。一方、オートスクロール操作はホイールボタンで行うため、文書選択との併用が可能である。オ

ートスクロールは、押した位置からの相対位置と距離によってスクロールの方向と速度が決まり、位置の微調整もページ間を一気に飛ばして移動することも可能となるが、スクロールの速度制御が難しい。また、ものによってはずっとドラッグしていなければならないものもあり（Adobe Readerがそうである）、手が疲れてしまう。

これらの問題は、クリック操作で移動する方法を用いると解決する。クリック移動方式を、図6を用いて説明していく。図6では、ページがビューア画面からはみ出ている状態となっているのだが、この画面の色のついた部分をクリックすると移動が開始される。この移動は8方向に分かれていて、縦と横のスクロールバーの値を調節することで制御される。制御はアニメーションを伴って行われるので直感的にわかりやすくなっている。図6で、ビューア画面右端をクリックすると、次のページへと移動する。このとき、水平スクロールバーの値のみを調節するため、上下の移動は行われない。ここでは、画面がページ下端にあるのを保ったまま、2ページの左端が画面左端にくるように移動が行われる。左端をクリックした場合は、この逆で前のページに移動する。画面上端をクリックすると、現在表示しているページの上端へと移動する。下端をクリックした場合はこの逆となる。濃い青の部分をクリックした場合の動作は、薄い青（左端、右端、上端、下端）の部分の動作の組み合わせとなる。例えば、図6で右上端



図5 Adobe readerの右クリックメニュー

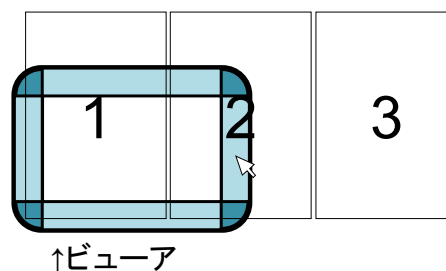


図6 クリック移動

をクリックした場合、2ページ目の上端へと画面が移動する。

提案方式によるクリック移動は、移動にドラッグ操作が必要なくなって文書選択との併用が可能であり、画面のどの部分をクリックするかで移動方向がわかるため直感的にもわかりやすい。また、ページを掴んでいる必要もなく難しい速度制御もなくなり、手にかかる操作の負担が減ると考えられる。

2-3 逆N字スクロール

ここまで横スクロールのよさを伝えてきたが、横スクロールにすることで新たな問題も生じる。従来方式だと、ページ内の移動もページ間の移動もどちらも縦で一致していた。そのため、読み進めていくときには、（ページが画面に収まっている限りは）垂直スクロールバーの値さえ変わっていってくればよい。つまり、マウスホイールを回すだけで読み進めていくことができる。しかし、横スクロールの場合はこうはいかない。ページ内の移動は垂直スクロール、ページ間の移動は水平スクロールと分かれてしまっているからである。これは、先ほどの図6の例を見てもらえるとわかりやすい。1ページを読む場合は上から下へとスクロールし、次のページへいく場合は左から右へとスクロールする。ここで、2ページ目の上へ移動したいときは、右にスクロールしてから上へスクロールするというふうに、少なくとも2回スクロール操作をしなければならない。2-2であげたクリック移動しか使わないのであれば、右上をクリックするだけでいいのだが、実際にはマウスホイールも使うはずである。だが、横スクロール方式でマウスホイールを用いると、ページ内の移動で終わってしまい、次のページへとってくれない。そこで、従来方式の利点を崩さないためにも新たな機能を追加する。それが、逆N字スクロールである。

逆N字スクロールでは図7のように、スクロール中に垂直スクロールのつまみが一番下まできたとき（①）、自動的に次のページの先頭に画面がくるように、垂直水平両スクロールの値を調節する（②③）。イメージ的には、図8のような移動を繰り返していくことになる。赤が自分でスクロールしていく部分で、黄が、システムが自動でやってくれる部分である。このように、Nを逆に描くように進んでいくため、逆N字スクロールと呼んでいる。クリック移動でもこのイメージでいけるように、画面がページ下端にあるときに画面下端をクリックすると次のページの上端（先頭）へ、画面がページ

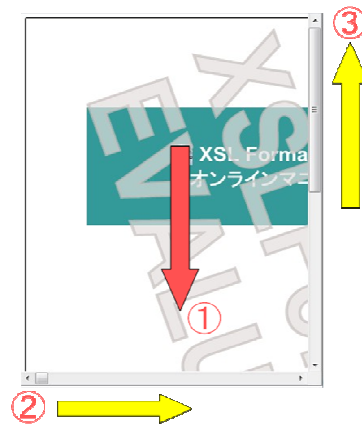


図7 逆N字スクロール

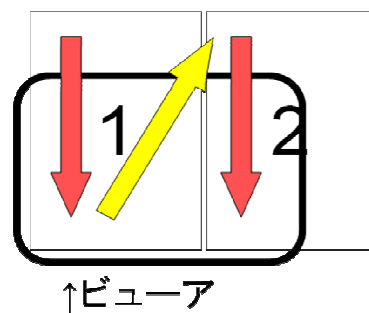


図8 逆N字スクロールのイメージ

上端にあるときに画面上端をクリックすると前のページの下端へと画面が移動するように実装する。例えば、図6の状態では画面下端をクリックすると2ページ目の上端へと画面が移動する。

この機能を実装することにより、提案方式の利点を保ったまま、従来方式の利点も失われずにすむようになる。

2-4 ページスライダー

従来方式だと、図9のようにページ番号を入力するボックスやスピンボックスがある。ここに、ページ番号を入力することで瞬時にそのページに移動できる。本ツールでは、これだけでなくスライダー式でページを指定できるものも実装する。それがページスライダーである。これを図10に示す。

ページスライダーは、動きが水平スクロールバーと同期しており、ページの先頭位置にあうように目盛りがふってあるので、現在どのページに画面があるかがより直感的にわかりやすい。また、ページを探すときに具体的なページ数を覚えていなくても、だいたいどのあたりにあったかという印象から探しやすくなっている。

図9 Adobe Readerにおけるページ番号入力ボックス

(1) Can-do	Each unit has two (for the unit are bef assess each student from zero to eight p Can-dos for Units 7.
(2) Participation	Students will be ass actively in class. Fs participant' and 'le awarded from zero b
(3) Preparation	At the start of each is related to the Prej or not the student: contents before con accordingly.
(4) Lesson Review Task	After each lesson, tl

図10 ページスライダー

このようなページや画像の一覧をスライダー形式で表示するといった機能は、Leeyes[3]のような連番画像ビューアでも用いられている。

3 開発言語

本研究では、PDFのレンダリングライブラリにpopplerを用いている。popplerはオープンソースのライブラリなので、開発に適している。また、他のオープンソースのものに比べて、ファイルの読み込む速度が比較的安定しているように感じられたので、これを用いていくことに決めた。開発環境はQtCreatorという、Qtを用いるのに特化した環境を使用した。QtはC++をGUI開発に特化させたアプリケーションフレームワークで、popplerとの相性も非常に良い。実際、popplerとQtの組み合わせで開発されたビューアにOkularというものがあるが、安定した動作を見せている。

4 実験および評価方法

提案方式の有効性を検証するため、被験者実験を行う。実験では、条件が同じになるように従来方式と提案方式で同じ文書を読ませるので、被験者の読む速さは比較対象にいけない。どうしても後に行う方式の方が有利になってしまうからである。また、被験者の主観でどちらの方が使いやすかったか、マウスやキーボードでどの程度操作する必要があったか(どの程度手間がかかったか)に着目し、それぞれの項目と総合的な部分で評価する。

実験項目としては以下のようなもの考えている。

1, 本文書のように、図や表がページをまたがっていて、これらの図や表を参照しながら文書を読み進める場合、従来方式と提案方式どちらのほうが読みやすいか

2, 目標のページはどちらのほうが探しやすいか

3, ファイル全体をざっと見渡したいとき、どちらのほうが手軽に見渡しやすいか

これらについて、横スクロール、クリック移動、ページスライダーがどう効果を発揮しているかを評価していく。

「参考文献」

[1] 王 允龍, 羽山徹彩, 國藤 進, 電子ブックリーダーにおける物理的ページング操作を可能にしたセンシング装置の開発 p.5

[2] 大きなディスプレイでPDFなどの文書を読むときは横スクロールが最強だと思うので実装してみた

<http://d.hatena.ne.jp/tmatsuu/20120325/1332667357>

[3] Leeyes
<http://www3.tokai.or.jp/boxes/leeyes/>